ISSN 0316-6295

C25.1

Bounding Algorithms for Queueing Network Models of Computer Systems

by

Derek L. Eager

Technical Report CSRG–156

March 1984

COMPUTER SYSTEMS RESEARCH GROUP
UNIVERSITY OF TORONTO

Bounding Algorithms for Queueing Network Models of Computer Systems

by

Derek L. Eager

Technical Report CSRG–156

March 1984

The Computer Systems Research Group (CSRG) is an interdisciplinary group formed to conduct research and development relevant to computer systems and their application. It is jointly administered by the Department of Electrical Engineering and the Department of Computer Science of the University of Toronto, and is supported in part by the Natural Sciences and Engineering Research Council of Canada.

© Copyright 1984, Computer Systems Research Group, University of Toronto.

Digitized by the Internet Archive
in 2018 with funding from
University of Toronto

https://archive.org/details/technicalreportc156univ

Bounding Algorithms for Queueing Network Models of Computer Systems

by

Derek L. Eager

A Thesis submitted in conformity with the
requirements for the degree of
Doctor of Philosophy

Department of Computer Science

University of Toronto

March 1984

© Derek L. Eager, 1984

ABSTRACT

This thesis is concerned with the problem of efficiently analyzing large separable queueing network models of computer systems. Obtaining the exact solution of such a network is often infeasible. Approximation techniques have been proposed that significantly reduce the cost of network analysis; however, these techniques lack bounds on the errors that they introduce. This thesis proposes algorithms that bound, rather than only estimate, the queueing network solution.

Each bounding algorithm is based on the recursive structure of one of the exact solution algorithms, in which a queueing network solution is computed from the solutions of simpler networks. In an exact solution algorithm, the recursion basis consists of trivial networks whose solutions are known a priori. In each of the proposed bounding algorithms, the recursion basis consists of much more complex networks, for which appropriate approximate solutions are found. In this way, the proposed bounding algorithms significantly reduce the cost of network analysis.

Each bounding algorithm is analyzed with respect to its computational cost and its accuracy. A comparison is given among proposed and previous bounding algorithms, with the goal of determining appropriate bounding algorithms for various accuracy requirements, cost constraints, and queueing network types.

ACKNOWLEDGEMENTS

I am indebted to my supervisor, Professor K.C. Sevcik, for the guidance and encouragement that he has given me. His influence on my graduate career and on this work has been great.

I wish to thank Professor G.S. Graham for his help in improving this thesis. Dr. S.S. Lavenberg, and Professors J.N.P. Hume, R.A. Mathon, A.O. Mendelzon, M.L. Molle, and J.G.C. Templeton also made many helpful comments and suggestions, for which I am grateful.

Many people have contributed to making my time at the University of Toronto enjoyable. I would like in particular to thank Ignacio Casas, Bruce Galler, and the other members of the "Project SAM" modelling group.

Finally, I thank my wife Marlene for her understanding and patience, and her help in preparing this thesis.

Generous financial support was provided by the Natural Sciences and Engineering Research Council of Canada.

TABLE OF CONTENTS

Abstract i

Acknowledgements ii

Table of Contents iii

1 Introduction 1

1.1 Performance Modelling 1

1.2 Queueing Network Models 2

1.3 Model Analysis 3

1.4 Thesis Outline 6

2 Separable Queueing Networks 8

2.1 The Class of Separable Queueing Networks 9

2.2 Solution Algorithms 15

2.3 Approximate Analysis Techniques 22

2.4 Bounding Algorithms 24

Appendix 2.A 32

3 The Single Closed Class Performance Bound Hierarchy Algorithm 34

3.1 The Basic Algorithm 35

3.2 An Algorithm Modification 48

3.3 Symbolic Formulation 54

3.4 Computational Cost 58

3.5 Algorithm Accuracy 60

3.6 Modifications to Allow Open Customer Classes 67

Appendix 3.A 72

Appendix 3.B 79

4 The Multiple Closed Class Performance Bound Hierarchy Algorithm **82**

4.1 Algorithm Definition 83

4.2 Symbolic Formulation 98

4.3 Computational Cost 100

4.4 Algorithm Accuracy 102

4.5 An Error-Bounded Approximation Algorithm 108

4.6 Modifications to Allow Load Dependence 112

Appendix 4.A 117

5 Looped Performance Bound Hierarchy Algorithms **123**

5.1 The Single Class aMVA/PE Algorithm 124

5.2 A Single Class Optimistic Looping Algorithm 131

5.3 A Conjectured Single Class Optimistic Looping Algorithm 139

5.4 A Looped Single Closed Class Performance Bound Hierarchy Algorithm 144

5.5 A Looped Multiple Closed Class Performance Bound Hierarchy Algorithm 160

Appendix 5.A 173

Appendix 5.B 186

6 Bounding Algorithms Based on Convolution **188**

6.1 Bounding Algorithms Based on Customer Recursion Convolution 189

6.2 Bounding Algorithms Based on Service Center Recursion Convolution 196

Appendix 6.A 203

7 A Comparison Among Bounding Algorithms **206**

7.1 Applicability 206

7.2 Implementation Effort 207

7.3 Accuracy and Computational Cost 208

7.4 Recommendations 212

8 Conclusions and Directions for Further Research **214**

8.1 Summary of Thesis Work 214

8.2 Directions for Further Research 217

Appendix A (Notation Summary) 219

References 223

CHAPTER 1

INTRODUCTION

Advances in technology have led to increasingly complex and powerful computer systems. At the same time, user expectations of the quantity and quality of the service provided have increased. These two trends make the tasks of system design, configuration planning, and system tuning ever more difficult and crucial.

These tasks cannot be effectively performed without the ability to determine the performance characteristics of both proposed and existing computer systems. For this purpose, queueing network models are important tools, and are the subject of this thesis. In particular, methods of efficiently analyzing queueing network models for the required performance predictions are investigated.

Section one of this introduction considers the motivation and the methodology for performance modelling. Section two describes the particular modelling tool of interest, the queueing network. The principal methods of analyzing queueing networks are treated in section three. Finally, section four outlines the thesis contents.

1.1. Performance Modelling

Knowledge of the performance characteristics of a computer system is required throughout the computer system life cycle: from system design to configuration planning to system tuning. Since the expense of system

modification for experimental purposes may be prohibitive, many of the performance questions that arise during the system life cycle cannot be answered by measurement. Even when the appropriate hardware and software are in place, often a projected rather than an existing workload is of interest. These factors motivate the use of *performance modelling*, in which an abstraction (model) of a system is used to derive system performance measure estimates.

The most common form of performance modelling involves three phases [Lazowska et al. 84]. In the *validation* phase, a model is constructed and its sufficiency established through a number of experiments in which the model is parameterized to reflect the attributes of an existing system. In each experiment, model performance measure values are compared to available measurement data. In the *projection* phase, the model is used to derive performance predictions for the system of interest (typically, a modification of an existing system). The *verification* phase serves to validate the model further, and takes place when measurement data for the system of interest becomes available.

In addition to its generality, modelling has the advantage of offering a spectrum of available accuracy and cost levels. Very simple models (based perhaps only on simple extrapolation) can be developed and utilized with very little associated cost. If the accuracy of these models is insufficient, it is possible to develop more complex and costly models that achieve greater accuracy. Practical experience has shown that queueing network models can achieve a suitable compromise between accuracy and cost [Graham 78].

1.2. Queueing Network Models

The amount of contention for resources is a critical factor in the performance of a computer system. A natural choice for a computer system model is therefore a network of queues, or *queueing network*. Each queue is associated

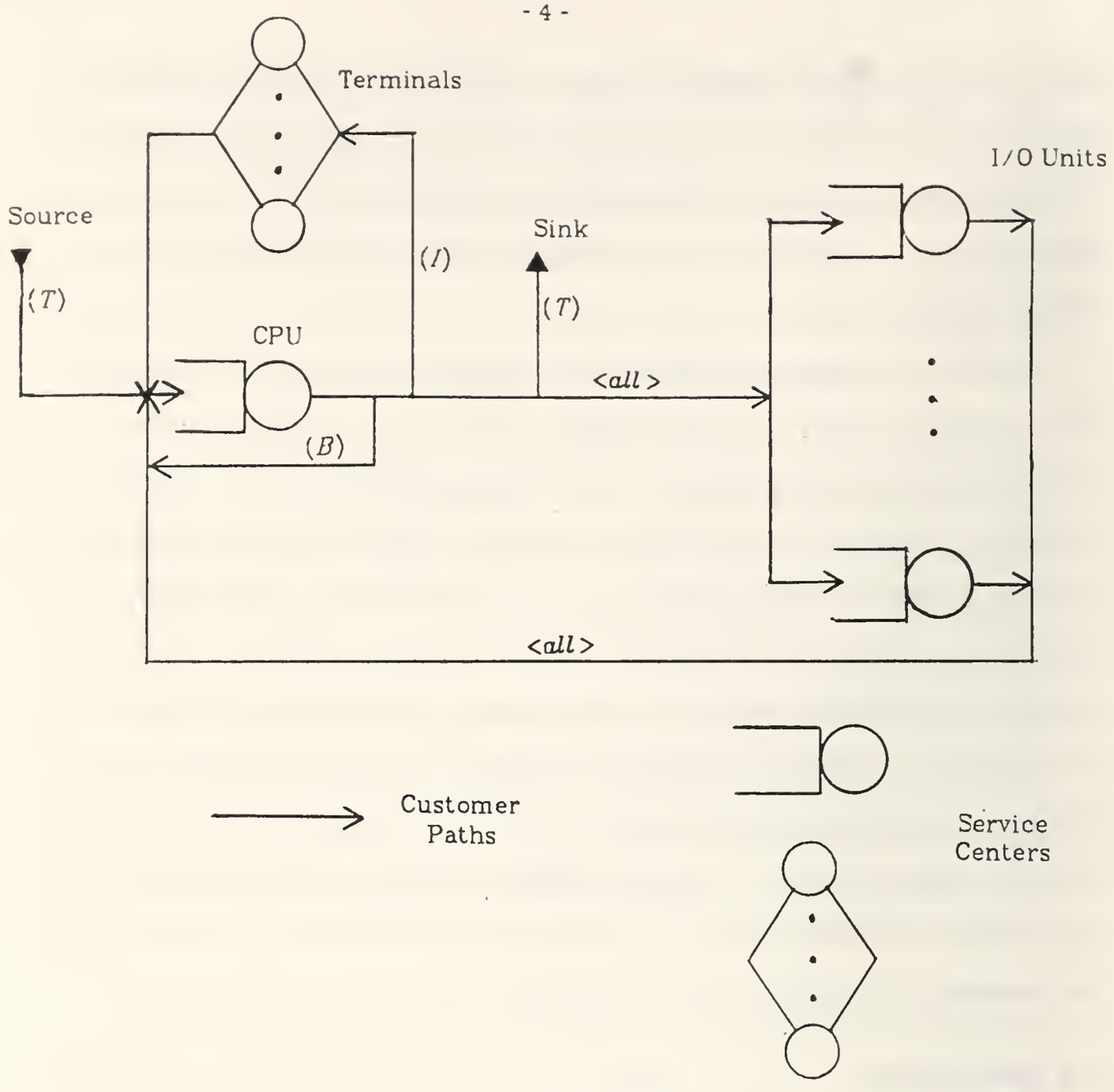
with a particular system resource, such as a disk unit or a processor. User work units, the *customers* of the queueing network, move from queue to queue as they demand and release system resources. Each queue has an associated *server* that provides service to its enqueued customers; a queue-server pair is termed a *service center*.

Modelling is based on the abstraction of essential characteristics from the mass of system details. A major example of this in the queueing network modelling context is the assumption of equivalence among queueing network customers. Customers are grouped into *classes*; customers within the same class are considered indistinguishable as far as their effect on system performance is concerned. As an example of the use of classes, a system with mixed interactive, transaction, and batch processing might require a class of "interactive" customers, a class of "transaction" customers, and a class of "batch" customers. An example queueing network model (unparameterized) of such a system is provided in figure 1.1. A special symbol is used for the service center representing the terminals since it is usually assumed that each user has his own terminal, and therefore never has to wait for service.

1.3. Model Analysis

Performance predictions are obtained from a parameterized model by the process of *model analysis*. In the queueing network modelling context, there is usually a direct correspondence between model "performance" measures, and actual system performance measures. For example, the utilization, by queueing network customers, of the service center representing a central processor can be used as an estimate of the actual central processor utilization.

Model analysis may be either *exact* or *approximate*. In an exact analysis of a queueing network model, exact values are obtained for the model performance measures. Any discrepancies between these values and the actual



Customer Classes

Class *B*: batch workload
Class *I*: interactive workload
Class *T*: transaction workload

An Example Queueing Network Model

Figure 1.1

system values are due to the approximations made in the model construction. Approximate analysis introduces additional error, since approximations rather than exact values are obtained for model performance measures. In many situations of practical interest, particularly when a large model must be analyzed, exact analysis is too expensive computationally to be of use, and approximate analysis becomes necessary.

It is important to distinguish between errors due to approximations in model construction and errors due to approximations in model analysis. The former errors decrease the credibility of the model, while the latter errors decrease the credibility of the model validation process itself. Only if there are no significant errors in model analysis can a modeller confidently assert that a model is or is not an accurate reflection of the system being modelled. It is shown in chapter two that the errors induced by at least one popular approximate analysis technique may be very significant, and in fact may completely dominate other errors.

When approximate analysis is used, confidence in the model validation process and in the performance predictions obtained from a model can be increased by providing the modeller with bounds on the maximum error that the approximate analysis could have induced. This can be done in three major ways. In a *bounding algorithm*, the approximate analysis technique used first computes bounds on model performance measures, and then using these bounds applies a simple heuristic (choosing bound interval midpoints, for example) to obtain point estimates. In an *error-bounded approximation algorithm*, on the other hand, bounds are computed only after a major portion of the computational effort has been expended on point estimates. Typically, the error bounds provided by an error-bounded approximation algorithm will be looser than those provided by a comparable bounding algorithm (for a given

computational investment), while the point estimates will be more accurate. *Hybrid* algorithms are also possible; in these algorithms significant computational effort is independently expended both on point estimates and on performance measure bounds. The computed point estimates may lie outside the computed bounds, in which case some heuristic is applied (discarding the computed point estimates and choosing appropriate bound interval endpoints, for example) to ensure consistency.

Approximate analysis techniques for queueing network models can be grouped into two major classes: analytic techniques and simulation. Since simulation is computationally expensive, often analytic techniques are preferred.

1.4. Thesis Outline

In this thesis, analytic techniques that provide bounds on induced errors are developed for the approximate analysis of queueing networks. The development is centered on analytic bounding algorithms; such an algorithm may then be used independently or as a component of a hybrid algorithm.

Many actual computer system modelling studies are based on *separable* queueing networks [Lazowska et al. 84] (also termed *product form* queueing networks [Baskett et al. 75] or *local balance* queueing networks [Sauer & Chandy 81]). Even when computer system characteristics require the use of a non-separable queueing network model, often the approximate model analysis requires analyzing several related, separable queueing networks. Chapter two characterizes the class of separable queueing networks and describes their existing exact and approximate analysis techniques. Previous applicable bounding algorithms are surveyed in detail, and the methodology that will be used to construct new bounding algorithms is outlined.

This methodology is based on the fact that the exact analysis techniques (*solution algorithms*) for separable queueing networks are recursive in structure, in that each computes the solution of a complex network from the solutions of simpler networks. The recursion basis used by each algorithm consists of queueing networks whose solutions are known a priori.

In the proposed methodology, a bounding algorithm is created by copying the computational structure of a solution algorithm. Computational cost is significantly reduced by utilizing a recursion basis that consists of queueing networks much "closer" to the desired network than those of the solution algorithm recursion basis. The solutions of these more complex queueing networks are not known a priori, and must be appropriately estimated.

There is a natural tradeoff between computational cost and accuracy in the bounding algorithms created using this methodology. By utilizing simpler queueing networks in the recursion basis, greater accuracy can be achieved at the cost of the additional computational steps that are then required.

In chapters three, four, five and six, this methodology is used to develop a number of distinct bounding algorithms. These bounding algorithms differ as to the type of separable queueing network treated, the solution algorithm used to provide the basic computational structure, and the precise manner in which the solution algorithm structure is utilized.

Chapter seven contains comparisons among two of the bounding algorithms introduced in previous chapters and a bounding algorithm that existed previously. Accuracy, computational cost, applicability, and implementation effort are the principal evaluation criteria that are considered. Recommendations are made as to appropriate bounding algorithms for various contexts.

Finally, chapter eight contains conclusions and directions for future research.

CHAPTER 2

SEPARABLE QUEUEING NETWORKS

In the computer system modelling context, separable queueing networks have two major advantages over non-separable queueing networks. First of all, an exact solution of a non-separable queueing network is usually computationally intractable. Separable queueing networks, on the other hand, can be solved relatively efficiently for the required performance measures. The computational expense required for an exact solution is prohibitive only when the queueing network is large. Secondly, separable queueing network models are relatively easy to parameterize, requiring only the workload intensity of each customer class and the *loadings* (measures of service demands) of customers of each class at each service center. Even these modest requirements strain the limits of available workload measurement and estimation tools.

In section one of this chapter, the class of separable queueing networks is characterized. This task is complicated somewhat by the existence of two fundamentally different views of queueing networks, those of *operational analysis* and *stochastic analysis* [Graham 78, Sevcik & Klawe 79]. Section two describes the major separable queueing network solution algorithms. Section three considers previous approximate analysis techniques; in particular, an example is given in which a popular approximate analysis technique induces an unacceptable error in queueing network performance measures. Finally, section four focuses on bounding algorithms. Previous bounding algorithms are surveyed, and the proposed methodology for constructing new bounding algorithms is

introduced. The chapter also has an associated appendix, appendix 2.A, which contains analytic results relevant to section three.

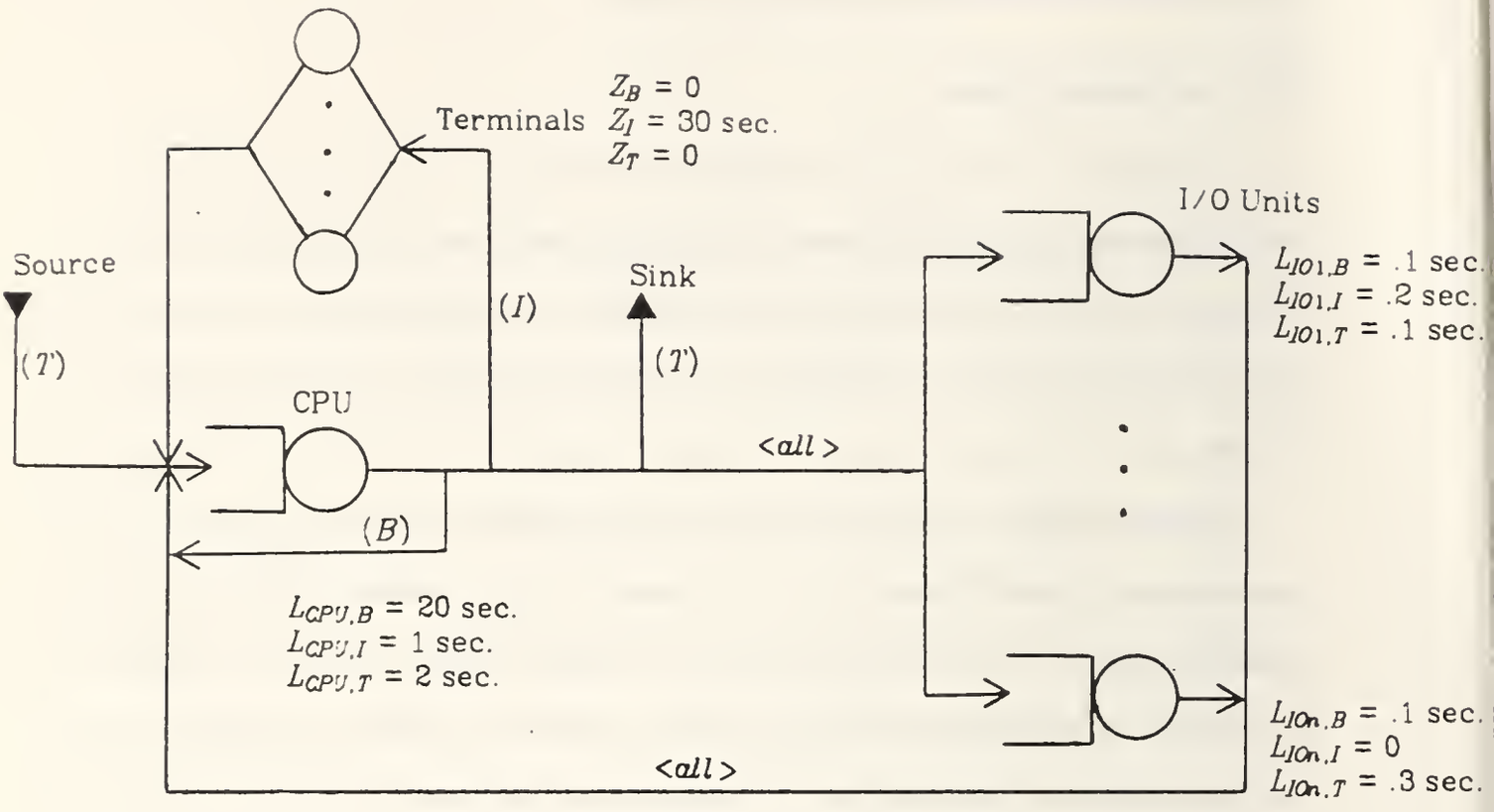
2.1. The Class of Separable Queueing Networks

2.1.1. An Example Model

In figure 2.1, the queueing network model of figure 1.1 is parameterized to yield a separable queueing network. This subsection discusses this parameterization, and in so doing introduces some of the basic notation and terminology required in this thesis. (A summary of all notation used is presented in appendix A.)

The transaction workload class has been specified as an *open* class. The customers of an open class r arrive at rate λ_r from a source external to the queueing network. The other classes are *closed* classes; there are a fixed number, N_s, of customers of a closed class s. The total number of customers in the closed classes is denoted by N. The complete workload intensity of a model is given by a pair $(\vec{N}, \vec{\lambda})$, where the vector $\vec{N}$ gives the populations of the closed classes, and the vector $\vec{\lambda}$ gives the arrival rates of the open classes. The open classes are indexed by the set O. If there is only one closed class, c is taken as its index; in general, the set C indexes the closed classes.

The queueing network service centers of the example model are of two types. The I/O units and the CPU are modelled by *fixed-rate* service centers, while the terminals are modelled by a single *delay* service center. At a fixed-rate service center, the total rate of service is constant regardless of the composition of the service center queue, assuming that at least one customer is present. At a delay service center, on the other hand, the rate of service afforded to each customer present is constant, regardless of the queue composition. As the number of queueing network customers present increases, so



Workload Intensities

Class B: N_B = 4
Class I: N_I = 40
Class T: λ_T = .1 transactions/second

A Parameterized Queueing Network Model

Figure 2.1

does the total service rate. Separable queueing networks may also include some types of *load-dependent* service centers, at which both the service rate received by each customer and the total service rate vary with the queue composition. Since it is trivial to obtain the performance measures of a queueing network with only delay service centers, it will be assumed that the customers of at least one class require service at a fixed-rate or load-dependent service center. The fixed-rate and load-dependent service centers will be indexed as service centers one through K; any delay service centers may be aggregated into a single delay service center, indexed as service center $K+1$, without any effect on queueing network performance measures.

Customer service requirements are specified through loadings. The service center k loading of a class s customer for some queue population $\vec{n}$ is defined here as the product of three factors: (1) the mean number of visits that a class s customer makes to center k, (2) the inverse of the center k throughput of class s customers, conditioned on a queue population of $\vec{n}$, and (3) the number of class s customers in the queue. For the service center types to be considered here, however, not all loadings need be explicitly specified. At an aggregate delay service center, the loading of a class s customer (denoted by Z_s) is constant for all possible queue populations. At a fixed-rate service center k, it is sufficient to specify the loading of a class s customer when no other customers are in the service center queue (denoted by L_{ks}). Finally, at one common type of load-dependent service center, the total service rate depends only on the total number of customers present. For such a service center k, it is sufficient to specify the loading of a class s customer for a total queue population of m (denoted by $L_{ks}(m)$), for all applicable m. For notational convenience, in the remainder of this thesis it will be assumed that the unit of time measurement chosen for class s is such that the sum of the class s fixed-

rate service center loadings equals one. (It is a characteristic of separable queueing networks that the performance measures of one class are independent of the time scale choice made for any other class.) However, this time scale choice is not necessary when applying the algorithms presented in this thesis.

Separable queueing networks require no additional parameters. The development of solution algorithms for these networks requires the use of a queueing network formalism; such a formalism also precisely defines which queueing networks, within the formalism, are separable. The operational and stochastic analysis formalisms are briefly considered in the next two subsections. Since both of these formalisms result in the same analytic expressions for the solutions of separable queueing networks, it is essentially unnecessary to assume either one or the other when considering these networks.

2.1.2. Operational Analysis

In operational analysis, a queueing network model is viewed as a representation of a system behavior sequence (which is itself an approximate representation of the system being modelled). Most work in operational analysis has been in a context of queueing networks with a single, closed customer class [Denning & Buzen 78, Sevcik & Klawe 79], although extensions are possible. For queueing networks with one or more closed customer classes, properties sufficient to ensure separability are as follows:

(1) *Flow balance*: The number of customer arrivals at a service center is the same as the number of customer departures.

(2) *One-step behavior*: No two "events" (customer service completions) occur at precisely the same time.

(3) *Routing homogeneity*: The destination of a customer when departing from a service center is independent of the distribution of the rest of the queueing network customers.

(4) *Device homogeneity*: The departure rate of class s customers from a service center k when a population of $\vec{n}$ is present at the service center is independent of the distribution of the customers not at the service center, and is given by $\frac{q_k(\vec{n}-\vec{e}_s)}{q_k(\vec{n})}$, where q_k is an arbitrary positive function and $\vec{e}_s$ denotes a population consisting of one class s customer. (Note that when there is only one customer class, arbitrary departure rates are allowed.)

It should be noted that the concept of the "service rate" of a service center (as was used in the previous subsection) is replaced in operational analysis by the concept of "departure rate". In general, the terminology used in this thesis will be independent of the underlying queueing network formalism, although occasionally it will be convenient to assume a context of stochastic analysis, which will be considered next.

2.1.3. Stochastic Analysis

In stochastic analysis, a queueing network model is viewed as a representation of a stochastic process (which is itself an approximate representation of the system being modelled). There are a number of different approaches within stochastic analysis to the problem of characterizing separable queueing networks. One approach is based on giving examples of allowable queueing network structures and service center *types* (types being fixed by customer service time demand distributions, service center scheduling disciplines and service rates) [Baskett et al. 75]. Other approaches give general properties that imply separability [Chandy, Howard & Towsley 77, Kelly 79, Noetzel 79, Sauer & Chandy 81, Chandy & Martin 83].

A common element to separability characterizations is the identification of service center types (by example or through general properties) that ensure that any queueing network appropriately constructed using only service centers of these types is separable. Typically (but not always [Towsley 80]) a queueing network construction is appropriate if customer routing among service centers is performed by Bernoulli trial with fixed routing probabilities, and if, for each open class, the customer arrival process is Poisson.

One approach to characterizing service center types that ensure separability is to consider a service center in *isolation*, at which customers arrive from an external Poisson source. Such a system has a state describable by a Markov process. A service center is of the type ensuring separability if, in isolation, the probability flux out of a state due to a customer of class s arriving is equal to the probability flux into that same state due to a customer of class s departing, for all classes s (for some choice of Markov process description) [Kelly 79].

This condition is satisfied by, for example, *processor-sharing* scheduling with general customer service time demand distributions, and *first-come-first-served* scheduling with exponential customer service time demand distributions that are identical for all customer classes [Barbour 76, Kelly 79]. At a processor-sharing service center, the total service rate is divided equally among all of the customers present. At a first-come-first-served service center, the total service rate is allocated to the customer at the head of the service center queue. Although the mean service time demand of a customer must be independent of the customer's class at this last type of service center, the mean number of visits may be class dependent. This implies that the loadings may also be class dependent.

2.2. Solution Algorithms

2.2.1. Performance Measures

A solution algorithm yields exact queueing network performance measures. There are a variety of performance measures appropriate to separable queueing networks that are of interest to the modeller; these differ with respect to their importance and their associated computational expense. In this thesis, performance measures are grouped into three classes.

The *primary* performance measures include the class s system throughput ($X_s(\vec{N},\vec{\lambda})$), class s mean system residence time ($R_s(\vec{N},\vec{\lambda})$), and the service center k utilization due to class s customers ($U_{ks}(\vec{N},\vec{\lambda})$). In the definition of mean system residence time, the mean residence time at an aggregate delay service center (equal to Z_s) is excluded, as motivated by the usual application of delay service centers to the modelling of terminals.

The *secondary* performance measures include the mean queue length of class s customers at service center k ($n_{ks}(\vec{N},\vec{\lambda})$) and the mean residence time, summed over all of the customer visits, of a class s customer at service center k ($R_{ks}(\vec{N},\vec{\lambda})$). These performance measures are of somewhat less importance to the modeller than are the primary performance measures, and may also require additional computational expense.

Finally, the *tertiary* performance measures include the service center queue length distributions and those performance measures (queue length variances, for example) that can be derived from these distributions. Although some information is lost, for clarity class distinctions will not be made here. The quantities of interest for service center k are therefore the probabilities $p_k(m \mid \vec{N},\vec{\lambda})$ of m customers enqueued at the service center, for all appropriate m. Tertiary performance measures are usually of interest to the modeller only

for special applications.

2.2.2. A Classification of Solution Algorithms

It is computationally trivial to obtain the performance measures of the previous section for a separable queueing network with only open customer classes (with constant arrival rates). It will therefore be assumed in the remainder of this thesis that there is at least one closed customer class. For these queueing networks, any open classes can be algebraically treated; significant computational expense is incurred only in the solution algorithm that must be invoked to solve a queueing network consisting only of the closed customer classes (a *closed* queueing network) [Zahorjan & Wong 81].

All of the practical solution algorithms for closed separable queueing networks are based on using the solutions for simpler networks to derive solutions for more complex networks. In an algorithm based on *service center recursion*, the solution of a complex network is derived from that of a simpler network with one fewer service center. The recursion basis is the known solution for a queueing network with only one service center. The other major form of recursion is *customer recursion*, in which the solution of a complex network is derived from that of simpler networks with fewer customers. Often, the simpler networks consist of one network, containing one fewer class s customer, for each class s. The recursion basis is the network with no customers.

Solution algorithms employing customer recursion may be further classified as either *symbolic* or *non-symbolic*. Symbolic solution algorithms are based on evaluating the equations resulting from symbolically performing the customer recursion of some other (non-symbolic) solution algorithm. The major computation of a symbolic algorithm produces only primary performance measures. Additional computational effort can yield secondary or tertiary performance measures for selected service centers; however, it is not cost-

effective to obtain these performance measures for all service centers.

Finally, solution algorithms may be classified as to whether or not *normalization constants* are computed. A queueing network normalization constant is the sum of the unnormalized *product form* queueing network state probabilities [Baskett et al. 75]. Normalization constants, for various queueing network compositions and populations, can be used to derive all of the performance measures of interest.

In table 2.1, the following solution algorithms are classified using the criteria developed in this section: the Algorithm to Coalesce Computation of Normalization Constants (CCNC) [Kobayashi 79, Chandy & Sauer 80], Convolution [Buzen 73, Reiser & Kobayashi 75, Lam & Lien 81,83, Lam 82], the Local Balance Algorithm for the Computation of Normalization Constants (LBANC) [Chandy & Sauer 80], Mean Value Analysis (MVA) [Reiser & Lavenberg 80, Sauer 83], the Normalized Convolution Algorithm (NCA) [Reiser 81], and symbolic MVA (to be introduced in this thesis). The Convolution, MVA, and symbolic MVA algorithms will be discussed briefly, assuming a closed queueing network context, in the following three sections.

	Service Center Recursion	Customer Recursion	
		Symbolic	Non-symbolic
No Normalization Constants	NCA	symbolic MVA	MVA
Normalization Constants	Convolution	CCNC	LBANC

A Classification of Solution Algorithms

Table 2.1

2.2.3. Convolution

Convolution was the earliest efficient solution algorithm to be developed. It is based on the computation of normalization constants from which the performance measures of interest can be derived. The population level $\vec{n}$ normalization constant for the queueing network consisting of service centers one through k will be denoted here by $G(\vec{n},k)$, and is computed using

$$G(\vec{n},k) = G(\vec{n},k-1)+\sum_{r\in C} L_{kr}\, G(\vec{n}-\vec{e}_r,k)$$

for a fixed-rate service center k,

$$G(\vec{n},k) = \sum_{\vec{m}\le\vec{n}} G(\vec{n}-\vec{m},k-1)\prod_{r\in C}\frac{\prod_{l=prec_r+1}^{prec_r+m_r} L_{kr}(l)}{m_r!}$$

for a load-dependent service center k at which the service rate is dependent only on the total number of queueing network customers enqueued (where $\vec{m}\le\vec{n}$ holds if and only if $m_r\le n_r$ for all classes r, and $prec_r = \sum_l m_l$ for l indexing the classes that precede class r in some arbitrary ordering of the elements of C). and

$$G(\vec{n},K+1) = \sum_{\vec{m}\le\vec{n}} G(\vec{n}-\vec{m},K)\prod_{r\in C}\frac{Z_r^{m_r}}{m_r!}$$

for an aggregate delay service center indexed as service center $K+1$. Convolution has been classified here as using service center recursion because of the usual order of applying these equations, in which $G(\vec{n},k-1)$ values for all necessary $\vec{n}$ are computed before any of the $G(\vec{n},k)$ values.

Zahorjan has performed a comprehensive analysis of the complexity of both Convolution and MVA [Zahorjan 80]. For our purposes, it is sufficient to note that the complexity precludes the exact solution of large queueing networks (in particular, those with many customer classes) in most cases. However, there are exceptions; "tree" Convolution and MVA algorithms have been

developed that allow substantial computational savings for queueing networks in which the customers of each class visit only a few service centers [Lam & Lien 81,83, Hoyme et al. 82].

Convolution as described here has a numerical stability problem, in that some of the intermediate normalization constant values may lie outside the available floating point range for any one choice of loading time units [Chandy & Sauer 80, Reiser 81]. Lam has essentially solved this problem in a modification of the Convolution algorithm in which loadings are re-scaled when necessary to ensure appropriate normalization constant magnitudes [Lam 82].

2.2.4. Mean Value Analysis

MVA has only recently been extended to handle the full range of separable queueing networks [Sauer 83]. Due to computational difficulties induced by the presence of load-dependent service centers, its primary application is to queueing networks with delay and fixed-rate service centers only. It is this context in which MVA will be described here, extensions will be developed later as required.

The most intuitive derivation of MVA is based on the Arrival Instant theorem [Sevcik & Mitrani 81, Lavenberg & Reiser 80, Buzen & Denning 80, Kelly 79], which states that, for separable queueing networks, the distribution of network states "seen" by a customer at arrival to a service center is the equilibrium distribution for the network with that customer removed. A consequence of this result is the following equation for the class s mean residence time at a fixed-rate service center k:

$$R_{ks}(\vec{N}) = L_{ks}(1+\sum_{r\in C} n_{kr}(\vec{N}-\vec{e}_s))$$

The class s mean system residence time is then given by the sum of the class s fixed-rate service center mean residence times:

$$R_s(\vec{N}) = \sum_{k=1}^{K} R_{ks}(\vec{N})$$

All other primary and secondary class s performance measures for population level $\vec{N}$ can be derived using Little's equation [Little 61] from the $R_{ks}(\vec{N})$ and $R_s(\vec{N})$ values. (Tertiary performance measures can be obtained only by utilizing the same computational procedures required to handle load dependence.) The primary and secondary performance measures for a population of $\vec{N}$ can therefore be derived from those for all feasible populations with one fewer customer. This provides the linkage needed for customer recursion.

2.2.5. Symbolic Mean Value Analysis

Symbolic MVA is based on evaluating the equations resulting from symbolically performing the MVA recursion. The characteristic of the symbolic MVA algorithm that makes it of interest is its partitioning of the computation into two major steps. In the first step, various products of service center loadings are computed. In the second step, the required performance measures are computed using the products of service center loadings found in the first step. (An imperfect, but perhaps conceptually useful analogy is between computing queueing network performance measures and evaluating a polynomial in a variable x. The queueing network loadings correspond to the variable x, MVA corresponds to Horner's rule, and symbolic MVA corresponds to the "direct" method of polynomial evaluation.) The structure of symbolic MVA can be used to advantage when the algorithm is used as a basis for a bounding algorithm.

The derivation of symbolic MVA starts from the following expression for $n_{kr}(\vec{N}-\vec{e}_s)$ (obtained by the use of Little's equation):

$$n_{kr}(\vec{N}-\vec{e}_s) = \frac{R_{kr}(\vec{N}-\vec{e}_s)}{Z_r+R_r(\vec{N}-\vec{e}_s)}(N_r-\delta_{rs})$$

Here $\delta_{rs}=1$ if $r=s$, and 0 otherwise. Substituting this expression into the MVA mean residence time equation yields

$$R_{ks}(\vec{N}) = L_{ks}\left(1+\sum_{r\in C}\frac{R_{kr}(\vec{N}-\vec{e}_s)}{Z_r+R_r(\vec{N}-\vec{e}_s)}(N_r-\delta_{rs})\right) \qquad (2\text{-}1)$$

or

$$R_{ks}(\vec{N}) = L_{ks} + \sum_{r \in C} \frac{L_{ks} R_{kr}(\vec{N}-\vec{e}_s)}{Z_r + R_r(\vec{N}-\vec{e}_s)}(N_r - \delta_{rs}) \qquad (2\text{-}2)$$

It is useful to define the *expansion ratio* at service center k for population $\vec{n}$ by

$$E_k(\vec{n}) = \frac{R_{kt}(\vec{n}+\vec{e}_t)}{L_{kt}}$$

where t denotes an arbitrary closed class such that $L_{kt} \neq 0$. This definition is unambiguous, since, from equation 2.1, the right-hand side is independent of the choice of t. For separable queueing networks, the ratio of residence time to loading depends only on the distribution of populations found at arrival instants, which is determined by the population $\vec{n}$.

Symbolically performing the MVA recursion leads naturally to $W(\vec{n},[s_1,\cdots,s_p])$ quantities, where $W(\vec{n},[s_1,\cdots,s_p])$ is defined by

$$W(\vec{n},[s_1,\cdots,s_p]) = \sum_{k=1}^{K}(\prod_{g=1}^{p} L_{ks_g})E_k(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g}) = \sum_{k=1}^{K}(\prod_{g\neq l} L_{ks_g})R_{ks_l}(\vec{n}-\sum_{g\neq l}\vec{e}_{s_g}) \qquad 1 \leq l \leq p$$

$W(\vec{n},[s_1,\cdots,s_p])$ is the weighted sum of the fixed-rate service center expansion ratios for a population with p fewer customers than $\vec{n}$ ($[s_1,\cdots,s_p]$ are the class identities of the missing customers), where the weights are the products of the service center loadings of the missing customers. Note that, by definition, $W(\vec{N},[s])$ is identical to $R_s(\vec{N})$.

Equation (2-2) yields the following equation defining $W(\vec{n},[s_1,\cdots,s_p])$ in terms of $W(\vec{n}_l,[r_1,\cdots,r_q])$ quantities such that $\vec{n}-\sum_{g=1}^{q}\vec{e}_{r_g}$ has one fewer customer than does $\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g}$:

$$W(\vec{n},[s_1,\cdots,s_p]) = L^{[s_1,\cdots,s_p]} + \sum_{r\in C}\frac{W(\vec{n},[s_1,\cdots,s_p,r])}{Z_r + W(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g},[r])}(n_r - \sum_{g=1}^{p}\delta_{rs_g}) \qquad (2\text{-}3)$$

where $L^{[s_1,\cdots,s_p]} = \sum_{k=1}^{K}\prod_{g=1}^{p} L_{ks_g}$. Symbolic MVA uses equation (2-3) to compute

$W(\vec{N},[s])$ for all $s \in C$ (starting from $W(\vec{n},[s_1,\cdots,s_p])$ quantities such that $n_r - \sum_{g=1}^{p} \delta_{rs_g} = 0$ for all classes r), yielding mean system residence times for all classes. Little's equation can then be used to derive the system throughput for each class and the service center utilizations due to each class.

The mean residence times (and therefore the mean queue lengths) for all classes at each fixed-rate service center k of interest can be computed using

$$W_k(\vec{n},[s_1,\cdots,s_p]) = (\prod_{g=1}^{p} L_{ks_g}) + \sum_{r \in C} \frac{W_k(\vec{n},[s_1,\cdots,s_p,r])}{Z_r + W(\vec{n} - \sum_{g=1}^{p} \vec{e}_{s_g},[r])} (n_r - \sum_{g=1}^{p} \delta_{rs_g}) \qquad (2\text{-}4)$$

where $W_k(\vec{n},[s_1,\cdots,s_p])$ is defined by

$$W_k(\vec{n},[s_1,\cdots,s_p]) = (\prod_{g=1}^{p} L_{ks_g}) E_k(\vec{n} - \sum_{g=1}^{p} \vec{e}_{s_g}) = (\prod_{g \neq l} L_{ks_g}) R_{ks_l}(\vec{n} - \sum_{g \neq l} \vec{e}_{s_g}) \quad 1 \leq l \leq p$$

The time complexity of symbolic MVA (in terms of the number of arithmetic operations required) is approximately $2K\prod_{r \in C}(N_r+1) + |C|\prod_{r \in C}(N_r+1)(N_r+2)$ (for primary performance measures only). The first term in this complexity represents the computation of the $L^{[s_1,\cdots,s_p]}$ quantities, while the second represents the recursive computation of the $W(\vec{N},[s])$ values using these quantities. This time complexity should be compared to that of MVA [Zahorjan 80], which is approximately $4K|C|\prod_{r \in C}(N_r+1)$. (The relationship between the space complexities of the two solution algorithms is similar.) Although symbolic MVA may not be a solution algorithm of choice, it will be seen that it is often suitable as a basis for a bounding algorithm.

2.3. Approximate Analysis Techniques

The most widely used of the approximate analysis techniques for closed separable queueing networks are the *approximate MVA algorithms* [Bard 79, Schweitzer 79, Reiser & Lavenberg 80, Zahorjan 80, Neuse & Chandy 81, Chandy & Neuse 82, Chow 83]. One of the most popular of these is termed here the

approximate MVA with Proportional Estimation (*aMVA/PE*) algorithm (as motivated by the fact that it is traditionally viewed as being based on a proportionality assumption). Proposed by Schweitzer [Schweitzer 79], this algorithm is an enhancement of an asymptotic algorithm developed earlier by Bard [Bard 78, 79] and Pittel [Pittel 76]. Bard has since used the algorithm extensively [Bard 80, 81].

As it will be viewed here, the aMVA/PE algorithm is based on the assumption that the mean service center and system residence times with one fewer customer are the same as those with the original population level. Using the form of the MVA mean residence time equation given by equation (2-1), this yields, for each class s,

$$R_{ks}(\vec{N}) = L_{ks}(1+\sum_{r\in C} \frac{R_{kr}(\vec{N})}{Z_r+R_r(\vec{N})}(N_r-\delta_{rs}))$$

Typically, the computation proceeds by making some set of initial guesses for the $R_{ks}(\vec{N})$ values, and then iterating until convergence is obtained.

One disadvantage of the approximate MVA algorithms is that they do not provide bounds on induced errors. For some of the more computationally expensive approximate MVA algorithms, such as Linearizer [Chandy & Neuse 82], large errors have never been observed. It is unknown if this is because large errors are not possible, or because exact solutions (and therefore the errors in approximate solutions) can typically be calculated only for queueing networks with a small number of classes.

For other approximate MVA algorithms, such as the aMVA/PE algorithm, there are known examples in which the error produced is very large. Figure 2.2 shows a queueing network for which the aMVA/PE algorithm induces unacceptable errors. In appendix 2.A, it is proved that, for $N_B=1$, $\varepsilon\to 0$ and $N_A\to\infty$, the relative error in the mean system residence time of class B tends to 100 per-

cent, while that in the system throughput of class B tends to 50 percent. These errors are unacceptable since they far exceed the expected error in the performance predictions of a validated queueing network model: 5 to 10 percent for throughputs and 25 to 50 percent for residence times [Lazowska et al. 84]. In table 2.2, the percent relative errors are shown for ε=.1 and various values of N_A.

An approximate analysis technique that is similar in philosophy to the bounding algorithms that will be proposed in this thesis was concurrently developed by Thomasian and Nadji for queueing networks with a single closed class [Thomasian & Nadji 81]. The algorithm offers a hierarchy of successively more accurate approximations, with the exact solution as its limit.

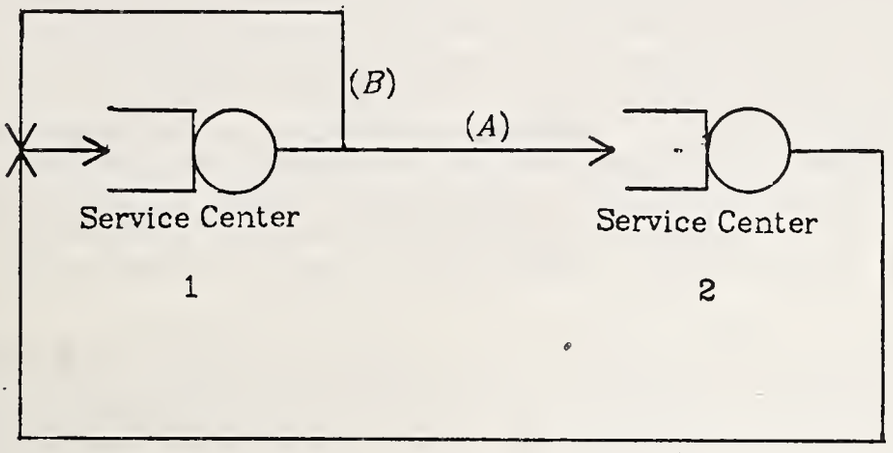
2.4. Bounding Algorithms

2.4.1. Previous Work

Asymptotic Bound Analysis (*ABA*) [Muntz & Wong 74, Kleinrock 76, Denning & Buzen 78] is the most general of the bounding algorithms currently available, and can be applied to some types of non-separable queueing networks as well as to separable queueing networks. Recall that the unit of time measurement chosen here is such that the sum of the fixed-rate service center loadings equals one. Then for queueing networks with a single closed customer class and only fixed-rate and delay service centers, the ABA bounds on mean system residence time are given by

$$max[NL_b - Z, 1] \leq R(N) \leq N$$

where b denotes the index of a *bottleneck* service center (a fixed-rate service center with a loading greater than or equal to that of all other fixed-rate service centers). Bounds on other primary performance measures can be derived with the use of Little's equation.



Loadings

	Service Center 1	Service Center 2
Class A	$\frac{1}{2+\varepsilon}$	$\frac{1+\varepsilon}{2+\varepsilon}$
Class B	1	0

Queueing Network for which the aMVA/PE Algorithm Induces an Unacceptable Error

Figure 2.2

N_A	percent relative error in $R_B(N_A,1)$	percent relative error in $X_B(N_A,1)$
10	33.1	24.9
20	40.4	28.8
40	50.2	33.4
100	71.0	41.5

Errors in an Application of the aMVA/PE Algorithm

Table 2.2

ABA bounds are quite loose due to the extreme assumptions made in deriving them. For example, the optimistic bound on mean system residence time (the lower bound) is derived by assuming that either the bottleneck service center is saturated, or no queueing is taking place. However, ABA bounds have proved useful in actual modelling studies [Sevcik, Graham & Zahorjan 80].

The *Balanced Job Bound* (*BJB*) algorithm [Zahorjan et al. 82] is less general than ABA, since it is applicable only to separable queueing networks, but it provides tighter bounds. The algorithm was originally developed for closed separable queueing networks consisting only of fixed-rate service centers. For these networks, the solution for class s performance measures has a very simple form if class s customers have the same loading at all service centers. For single class queueing networks, setting this loading to the average or the maximum of the actual loadings, while preserving the total loading by adding or removing service centers, yields optimistic or pessimistic bounds on primary performance measures. (Conceptually, "fractional" service centers may be needed when preserving the total loading; this does not affect the validity of the algorithm.) For multiple class queueing networks, the minimum rather than the average loading of each class must be used when deriving optimistic bounds; pessimistic bounds can be derived as for single class queueing networks.

Kriz has extended the BJB algorithm to treat queueing networks with delay service centers, and has also proposed an additional bound that is often tighter than one of the BJB bounds [Kriz 82].

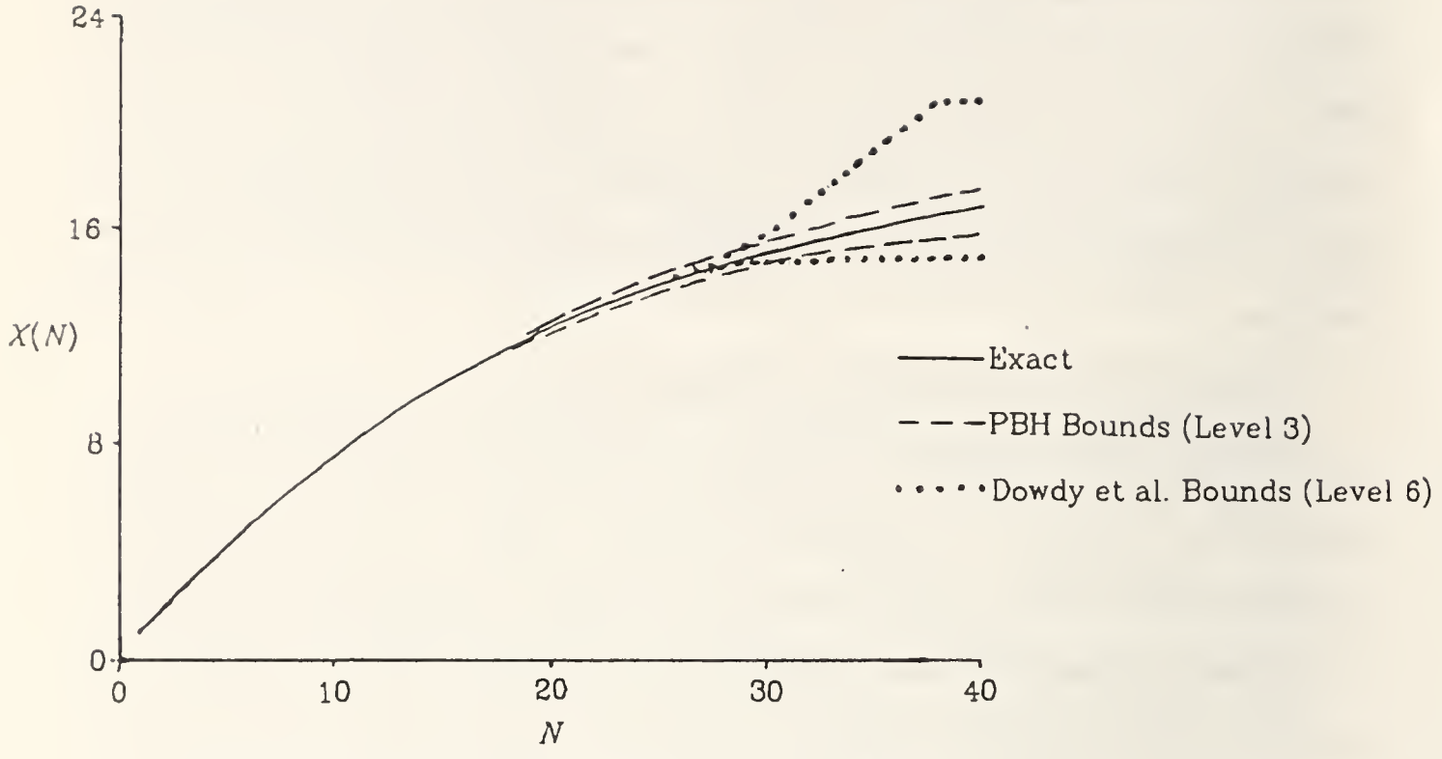
ABA and the BJB algorithm consider only the primary performance measures. Graham has developed bounds for secondary performance measures as well, in part with the use of ABA and BJB bounds [Graham 81].

The bounding algorithms that have been described above have the

disadvantage of providing only a limited tradeoff between accuracy and computational effort, and are usually unsuitable if the modeller requires high accuracy. (On the other hand, they are ideal if the modeller wishes to make only a minimal computational investment.) The remaining three bounding algorithms that will be surveyed here provide a significant tradeoff, but have major disadvantages.

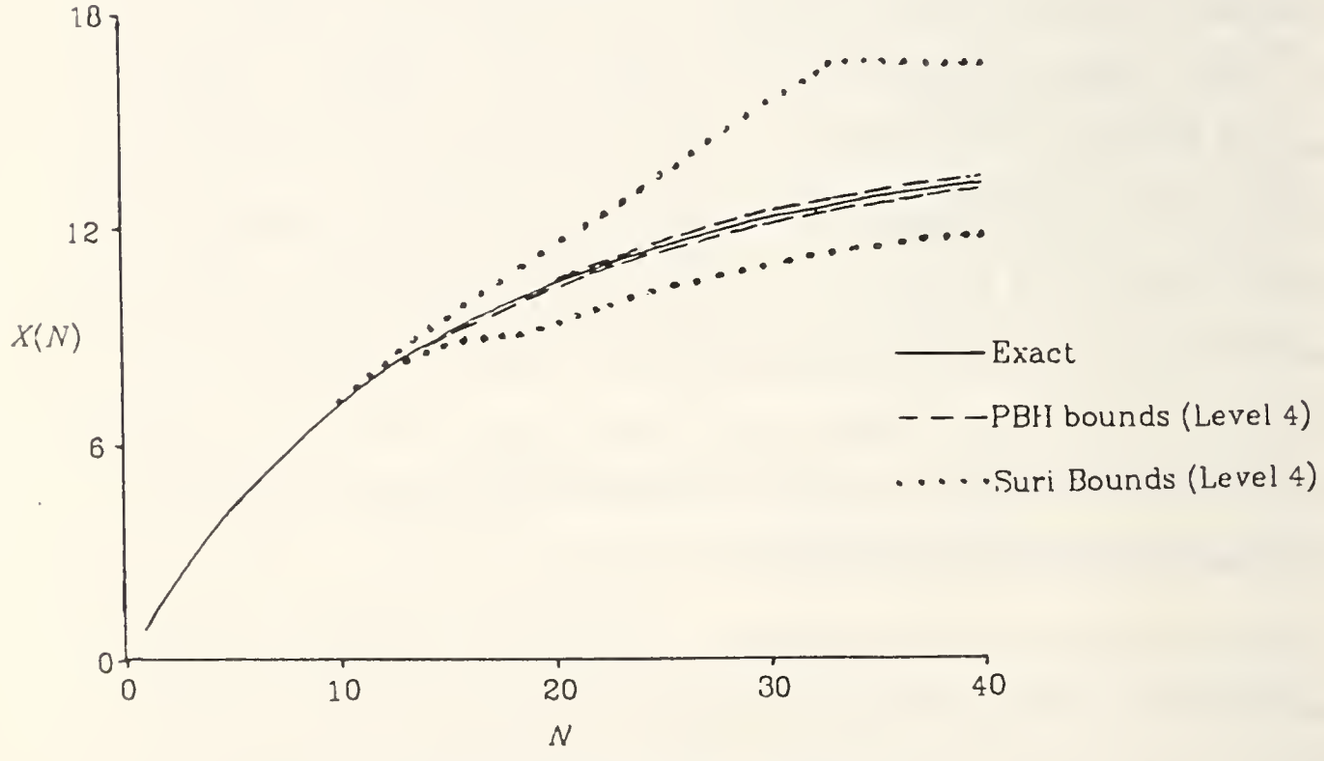
Dowdy, Perez-Davila and Stephens have developed an algorithm based on curve fitting that is applicable to separable queueing networks with a single closed class, and that is conjectured to produce bounds on primary performance measures [Dowdy, Perez-Davila & Stephens 83]. The algorithm proceeds by calculating queueing network performance measures for population levels 1 to n (for some choice of n), and then extrapolating to obtain bounds on the performance measures for the population level of interest. The major disadvantage of the algorithm is that it appears to yield looser bounds than alternative techniques for large customer populations. This is illustrated in figure 2.3, which shows system throughput bounds yielded by the Performance Bound Hierarchy algorithm (to be introduced in chapter three), and by the algorithm of Dowdy et al., for the queueing network used as an example by Dowdy et al. [Dowdy, Perez-Davila & Stephens 83]. The example queueing network has 50 fixed-rate service centers with loadings as follows: 1 center with loading 20/417, 2 with 19/417, 5 with 18/417, 5 with 15/417, 5 with 10/417, 8 with 7/417, 8 with 5/417, 8 with 4/417, and 8 with 2/417. The bounds shown for the two algorithms have similar computational costs.

Suri has developed a bounding algorithm applicable to separable queueing networks with a single closed customer class and only fixed-rate service centers [Suri 83]. The algorithm is quite similar in form to the symbolic formulation of the single closed class Performance Bound Hierarchy algorithm, but has a



PBH Bounds and the Bounds Proposed by Dowdy et al.
for an Example Queueing Network

Figure 2.3



PBH Bounds and the Bounds Proposed by Suri
for an Example Queueing Network

Figure 2.4

number of disadvantages with respect to this algorithm. Most importantly, for a given computational cost the bounds computed are significantly looser. This is illustrated in figure 2.4, which shows system throughput bounds yielded by the Performance Bound Hierarchy algorithm, and by the algorithm of Suri, for the queueing network used as an example by Suri [Suri 83]. The example queueing network has 49 fixed-rate service centers with normalized loadings as follows: 1 center with loading 21/343, 10 with 20/343, 4 with 10/343, 4 with 5/343, 3 with 4/343, 11 with 3/343, 1 with 2/343, and 15 with 1/343. The bounds shown for the two algorithms have similar computational costs.

The *PANACEA* bounding algorithm [McKenna & Mitra 81, McKenna, Mitra & Ramakrishnan 81, Ramakrishnan & Mitra 82a, 82b], developed at Bell Labs and used in the analysis of communication networks, is applicable to closed separable queueing networks with fixed-rate and delay service centers. The algorithm is based on the computation of terms in an asymptotic expansion of the multiple integrals with which normalization constants can be expressed. Two of the many restrictions imposed by the algorithm are that all classes must visit a delay service center, and that $\frac{L_{ks}}{Z_s}$ must be on the order of $\frac{1}{N_s}$ for all fixed-rate service centers k and classes s. PANACEA will be considered in more detail in chapter seven.

2.4.2. The Proposed Methodology

The current work is motivated by the need for generally applicable bounding algorithms that provide tradeoffs between accuracy and computational cost. It is proposed that a bounding algorithm be created by utilizing the recursive structure of a solution algorithm. Computational cost is reduced by utilizing a recursion basis that consists of one or more queueing networks that are more complex than the solution algorithm recursion basis networks (which are

trivial in the sense of having either no customers or only a single service center).

Since these new recursion basis queueing networks are more complex, their performance measures are not known a priori and must therefore be estimated. By choosing appropriate estimates (termed here the *initial estimates*), one attempts to ensure that the computation will yield optimistic or pessimistic bounds. This is feasible if the optimistic or pessimistic character of the initial estimates is preserved by the recursion equations used by the solution algorithm. It may not be possible to find such estimates without modifying these equations.

There is a natural hierarchy of bounds associated with a bounding algorithm created using this methodology. The position of a bound in the hierarchy is determined by the complexity of the recursion basis queueing networks that are used in deriving it. *Level* 0 bounds are derived by choosing as the recursion basis queueing network the network for which performance measure bounds are required; these bounds correspond to those used in forming the initial estimates. In general, *level* i bounds are derived by, depending on the solution algorithm, either choosing as the recursion basis a queueing network with i fewer service centers, or by choosing as the recursion basis all networks with i fewer customers.

In the next two chapters, the *Performance Bound Hierarchy* (*PBH*) algorithm will be developed by applying the methodology presented in this section to the MVA and symbolic MVA solution algorithms. Chapter three considers the PBH algorithm for queueing networks with one closed class and possibly some number of open classes, and consisting of fixed-rate and delay service centers only. Chapter four extends the algorithm to treat queueing networks with multiple closed classes and optionally some number of load-dependent service

centers. (Single closed class queueing networks that contain load-dependent service centers can be treated by specializing the multiple closed class algorithm.)

APPENDIX 2.A

This appendix considers the application of the aMVA/PE algorithm to the queueing network in figure 2.2. It is proved that, for $N_B=1$, $\varepsilon \to 0$ and $N_A \to \infty$, the relative error in the mean system residence time of class B tends to 100 percent, while that for the system throughput of class B tends to 50 percent.

From the MVA mean residence time equation, and since $N_B=1$,

$$R_B(\vec{N}) = 1+n_{1A}(\vec{N}-\vec{e}_B) \tag{2.A-1}$$

Since $L_{2A} = \frac{1+\varepsilon}{2+\varepsilon} > \frac{1}{2+\varepsilon} = L_{1A}$, for $N_A \to \infty$

$$n_{1A}(\vec{N}-\vec{e}_B) \to \frac{\frac{1}{1+\varepsilon}}{1-\frac{1}{1+\varepsilon}} = \frac{1}{\varepsilon}$$

Substitution in equation (2.A-1) yields

$$R_B(\vec{N}) = 1+\frac{1}{\varepsilon} \tag{2.A-2}$$

Superscripting quantities yielded by the aMVA/PE algorithm by $aMVA$, applications of Little's equation yield

$$n_{1A}^{aMVA}(\vec{N}) = \frac{R_{1A}^{aMVA}(\vec{N})}{R_A^{aMVA}(\vec{N})} N_A$$

and

$$n_{2A}^{aMVA}(\vec{N}) = \frac{R_{2A}^{aMVA}(\vec{N})}{R_A^{aMVA}(\vec{N})} N_A$$

From the algorithm definition and the previous two equations,

$$R_B^{aMVA}(\vec{N}) = 1+n_{1A}^{aMVA}(\vec{N}) \tag{2.A-3}$$

$$R_{1A}^{aMVA}(\vec{N}) = \frac{1}{2+\varepsilon}\left(2+n_{1A}^{aMVA}(\vec{N})\frac{N_A-1}{N_A}\right)$$

and

$$R_{2A}^{aMVA}(\vec{N}) = \frac{1+\varepsilon}{2+\varepsilon}\left(1+n_{2A}^{aMVA}(\vec{N})\frac{N_A-1}{N_A}\right)$$

Since these last two equations are asymptotically exact as $N_A \to \infty$, $n_{1A}^{aMVA}(\vec{N}) \to n_{1A}(\vec{N})$. Also, since $L_{2A} = \frac{1+\varepsilon}{2+\varepsilon} > L_{1A}$, as $N_A \to \infty$ $X_A(\vec{N}) \to \frac{2+\varepsilon}{1+\varepsilon}$. Therefore, as $N_A \to \infty$

$$R_{1A}(\vec{N}) \to \frac{1}{2+\varepsilon}\left(2+n_{1A}(\vec{N})\frac{N_A-1}{N_A}\right)$$

$$\Rightarrow X_A(\vec{N})R_{1A}(\vec{N}) \to X_A(\vec{N})\frac{1}{2+\varepsilon}\left(2+n_{1A}(\vec{N})\frac{N_A-1}{N_A}\right)$$

$$\Rightarrow n_{1A}(\vec{N}) \to \frac{1}{1+\varepsilon}\left(2+n_{1A}(\vec{N})\frac{N_A-1}{N_A}\right)$$

$$\Rightarrow n_{1A}(\vec{N}) \to \frac{2}{\varepsilon}$$

which yields

$$n_{1A}^{aMVA}(\vec{N}) \to \frac{2}{\varepsilon}$$

Using this last relationship to substitute into equation (2.A-3) produces

$$R_B^{aMVA}(\vec{N}) = 1+\frac{2}{\varepsilon}$$

The relative error in the mean system residence time of class B is therefore given by

$$\frac{(1+\frac{2}{\varepsilon})-(1+\frac{1}{\varepsilon})}{1+\frac{1}{\varepsilon}} 100\% = \frac{1}{1+\varepsilon} 100\%$$

This quantity tends to 100 percent as $\varepsilon \to 0$. The relative error in the system throughput of class B is given by

$$\frac{\frac{1}{1+\frac{1}{\varepsilon}} - \frac{1}{1+\frac{2}{\varepsilon}}}{\frac{1}{1+\frac{1}{\varepsilon}}} 100\% = \frac{1}{2+\varepsilon} 100\%$$

This quantity tends to 50 percent as $\varepsilon \to 0$.

CHAPTER 3

THE SINGLE CLOSED CLASS PERFORMANCE BOUND HIERARCHY ALGORITHM*

The computational expense of an exact solution of a separable queueing network can be prohibitive only when the network has multiple closed customer classes. However, there are still two major motivations for considering bounding algorithms for queueing networks with only a single closed class. First of all, in optimization studies (such as that of Bryant and Agre [Bryant & Agre 81]), or when utilizing approximate solution techniques (such as that of Zahorjan, Lazowska and Garner [Zahorjan, Lazowska & Garner 83]), it may be necessary to analyze a large number of networks, magnifying the effect of any computational savings. Secondly, bounding algorithms for queueing networks with a single closed class are of theoretical interest. In particular, they provide insight that can often be applied to the multiple closed class context.

This chapter develops the Performance Bound Hierarchy (PBH) bounding algorithm for queueing networks with a single closed customer class and possibly some number of open classes, and consisting of fixed-rate and delay service centers only. The basic PBH algorithm, for queueing networks with no open classes, is presented in section one. An algorithm modification that guarantees nesting of the bounds yielded by the algorithm is presented in section two. Section three presents the symbolic formulation of the algorithm that results when symbolic MVA, rather than MVA, is used to provide the algorithm

* Portions of the material presented in this chapter appear in the paper "Performance Bound Hierarchies for Queueing Networks" [Eager & Sevcik 83a].

structure. In section four, the computational cost of the PBH algorithm is determined for both the symbolic and the non-symbolic formulations. The accuracy of the algorithm is treated in section five. Finally, in section six the PBH algorithm is extended to allow for any number of open customer classes. The chapter also has two associated appendices. The lemmas that are required in the body of the chapter are proved in appendix 3.A, while appendix 3.B contains derivations of analytic results relevant to section five.

3.1. The Basic Algorithm

3.1.1. Algorithm Structure

The non-symbolic formulation of the PBH algorithm is derived by applying the methodology introduced in the last chapter to the MVA solution algorithm. For queueing networks of the type being considered, the MVA mean residence time equation for service center k $(1 \le k \le K)$ becomes

$$R_k(N) = L_k(1+n_k(N-1)) \tag{3-1}$$

From Little's equation,

$$n_k(N) = \frac{R_k(N)}{Z+R(N)} N \tag{3-2}$$

where, by definition,

$$R(N) = \sum_{k=1}^{K} R_k(N) \tag{3-3}$$

Level i PBH bounds for population level N, $i \le N$, are computed by choosing appropriate estimates of the performance measures with $N-i$ customers, and then applying the MVA equations until population level N is attained. If $i=N$, the computation is identical to that of MVA. For $i<N$, the computational cost is reduced, but approximate rather than exact performance measures are produced. Superscripting performance measure estimates resulting from i applications of the MVA equations by (i), the PBH algorithm applies equations (3-1)

to (3-3) to yield, for $i \geq 1$,

$$R_k^{(i)}(N) = L_k(1+n_k^{(i-1)}(N-1)) \tag{3-4}$$

$$n_k^{(i)}(N) = \frac{R_k^{(i)}(N)}{Z+R^{(i)}(N)} N \tag{3-5}$$

and

$$R^{(i)}(N) = \sum_{k=1}^{K} R_k^{(i)}(N) \tag{3-6}$$

Optimistic bounds on the primary performance measures require "optimistic" (under a suitable definition) initial estimates for the population level $N-i$ performance measures (in particular, the fixed-rate service center mean queue lengths), while pessimistic bounds require "pessimistic" initial estimates. Therefore, to obtain both upper and lower bounds on primary performance measures, two separate recursive computations must be performed, an *optimistic computation* (in which the optimistic initial estimates are utilized) and a *pessimistic computation* (in which the pessimistic initial estimates are utilized). As will be seen, both computations are needed to produce upper and/or lower bounds on secondary performance measures. Quantities produced during an optimistic or pessimistic computation will be subscripted by *opt* or *pess*, respectively. The problem of using the computed bounds to obtain suitable point estimates will be deferred until section 3.5.

3.1.2. The Initial Estimates

This subsection presents one choice of appropriate optimistic and pessimistic initial performance measure estimates. In an actual algorithm implementation, however, it is more computationally efficient to compute initial estimates from expressions for the performance measures of level one (those resulting from an application of the MVA equations to the initial estimates given here). This point will be discussed further when the computational cost of the algorithm is considered in section 3.4.

The chosen optimistic initial performance measure estimates (stated for population level N) use the optimistic ABA bound on mean system residence time denoted here by $a(N)$:

$$R_{opt}^{(0)}(N) = a(N) = max[NL_b - Z, 1]$$

An optimistic allocation of $R_{opt}^{(0)}(N)$ across the fixed-rate service centers is given by

$$R_{k\,opt}^{(0)}(N) = \frac{1}{K}\,a(N) \qquad 1 \le k \le K$$

This allocation is optimistic since it predicts more balanced mean service center residence times than actually occur. Finally, Little's equation yields

$$n_{k\,opt}^{(0)}(N) = \frac{1}{K}\,\frac{a(N)}{Z+a(N)}\,N \qquad 1 \le k \le K \tag{3-7}$$

The chosen pessimistic initial estimates use a pessimistic bound on mean system residence time denoted by $b(N)$:

$$R_{pess}^{(0)}(N) = b(N) = 1 + \frac{(N-1)L_b - (Z+1) + \sqrt{((N-1)L_b - (Z+1))^2 + 4(N-1)L_b}}{2}$$

The bound $b(N)$ is the positive root of the equation

$$b(N) = 1 + \frac{b(N)}{Z+b(N)}(N-1)L_b$$

and can be viewed as an extension to queueing networks with delay service centers of a pessimistic Balanced Job Bound algorithm bound [Kriz 82]. A proof that $b(N)$ is pessimistic is given in lemma 3.1 in appendix 3.A.

A pessimistic allocation of $R_{pess}^{(0)}(N)$ across the fixed-rate service centers is given by

$$R_{k\,pess}^{(0)}(N) = \begin{cases} b(N) & \text{for } k = b \\ 0 & \text{for } k \ne b, 1 \le k \le K \end{cases}$$

This allocation is pessimistic since it predicts more skewed mean service center residence times than actually occur. (It is assumed, for clarity, that there is only one bottleneck service center; the algorithm can be improved by utilizing information about multiple bottleneck service centers, but this will

not be done here.) Little's equation then yields

$$n_k^{(c)}{}_{pess}(N) = \begin{cases} \dfrac{b(N)}{Z+b(N)} N & \text{for } k=b \\ 0 & \text{for } k \neq b, 1 \leq k \leq K \end{cases} \tag{3-8}$$

Tighter bounds could be used in place of $a(N)$ and $b(N)$ in forming the initial performance measure estimates. However, $a(N)$ and $b(N)$ appear to be the most analytically and computationally simple bounds that are exact (asymptotically) for queueing networks with no queueing ($N=1$ and/or very large Z), and for queueing networks that are saturated (the utilization of the bottleneck service center approaches one). In chapter five, algorithms for computing much more accurate (but also more costly) initial performance measure estimates will be developed.

3.1.3. Majorization

The initial estimates of the previous subsection were chosen to ensure that their optimistic or pessimistic character is preserved by the MVA equations, in the sense that applying these equations i times, for any i, results in another set of estimates (for a population with i additional customers) with the same optimistic or pessimistic character. A formal definition of what is meant here by "optimistic" and "pessimistic" initial estimates requires the concept of queue length *majorization*. As it will be used here, this concept is similar to, but more restrictive than, weak majorization as developed by Marshall and Olkin [Marshall & Olkin 79]. Consider vectors of mean queue length values $(n_1^{v'}(N), \cdots, n_K^{v'}(N))$ and $(n_1^{v}(N), \cdots, n_K^{v}(N))$, where the fixed-rate service centers are indexed so that $L_k \geq L_{k+1}$ for $1 \leq k \leq K-1$. The weak majorization of Marshall and Olkin holds if $\sum_{k=1}^{j} n_k^{v}(N) \geq \sum_{k=1}^{j} n_k^{v'}(N)$ for $1 \leq j \leq K$. With majorization as it will be used here, $(n_1^{v'}(N), \cdots, n_K^{v'}(N))$ is said to be majorized by, or optimistic with respect to, $(n_1^{v}(N), \cdots, n_K^{v}(N))$ if

$$n_j^{v'}(N) \geq n_j^v(N) \frac{\sum_{k=1}^{j} n_k^{v'}(N)}{\sum_{k=1}^{j} n_k^v(N)} \qquad 1 \leq j \leq K$$

and

$$\sum_{k=1}^{K} n_k^v(N) \geq \sum_{k=1}^{K} n_k^{v'}(N)$$

(Only mean queue lengths $n_k^v(N)$ such that $n_1^v(N)>0$ and $n_k^v(N) \geq n_{k+1}^v(N) \geq 0$ for $1 \leq k \leq K-1$ will be considered; one implication of this restriction is that division by $\sum_{k=1}^{j} n_k^v(N)$ is always well-defined. Also, each of the $n_k^{v'}(N)$ values ($1 \leq k \leq K$) will be restricted to be greater than -1; this ensures that no difficulties arise when applying the MVA equations.)

Intuitively, the $n_k^{v'}(N)$ mean queue lengths are optimistic with respect to the $n_k^v(N)$ values since they: (1) predict less congestion, in total, at the fixed-rate service centers, and (2) predict relatively less congestion at the heavily loaded fixed-rate service centers and relatively greater congestion at the lightly loaded fixed-rate service centers. This second point can be verified most easily when $\sum_{k=1}^{j} n_k^{v'}(N)>0$. In this case, in each subsystem consisting of service centers one through j ($1 \leq j \leq K$), the optimistic mean queue lengths allocate a greater proportion of the total subsystem mean queue length to the service center with the smallest loading. (As can be seen by dividing both sides of the first majorization relation by $\sum_{k=1}^{j} n_k^{v'}(N)$.)

One type of queue length majorization is particularly strong, and is termed here *bottleneck* majorization. A vector of mean queue length values $(n_1^{v'}(N), \cdots, n_K^{v'}(N))$ is bottleneck majorized by another vector $(n_1^v(N), \cdots, n_K^v(N))$ if

$$n_b^{v'}(N) \leq n_b^v(N)$$

$$n_k^{v'}(N) \geq n_k^{v}(N) \qquad k \neq b$$

and

$$\sum_{k=1}^{K} n_k^{v}(N) \geq \sum_{k=1}^{K} n_k^{v'}(N)$$

Note that bottleneck majorization implies majorization, but that the reverse implication does not hold

Proofs that queue length majorization and bottleneck majorization are preserved by the MVA equations are given in lemmas 3.3 and 3.4 in appendix 3.A. These lemmas show that if some vector of mean queue length values is majorized or bottleneck majorized by some other vector, then applying the MVA equations i times to both vectors yields mean queue length vectors (for a population with i additional customers) for which the same relationship holds.

The following theorem uses the preservation of majorization to establish a relationship between the mean queue lengths computed during the optimistic computation of the PBH algorithm, and the exact mean queue lengths.

Theorem 3.1:

$(n_1^{(i)}{}_{opt}(N), \cdots, n_K^{(i)}{}_{opt}(N))$ is majorized by $(n_1(N), \cdots, n_K(N))$ for $N \geq i \geq 0$, $N \geq 1$.

Proof:

The theorem trivially holds for $i=N$. Consider $i<N$. From lemma 3.3, it need only be shown that $(n_1^{(0)}{}_{opt}(N-i), \cdots, n_K^{(0)}{}_{opt}(N-i))$ is majorized by $(n_1(N-i), \cdots, n_K(N-i))$.

Since $R(N-i) \geq a(N-i)$,

$$\frac{R(N-i)}{Z+R(N-i)}(N-i) \geq \frac{a(N-i)}{Z+a(N-i)}(N-i)$$

which yields, from equation (3-7) and Little's equation,

$$\sum_{k=1}^{K} n_k(N-i) \geq \sum_{k=1}^{K} n_k^{(0)}{}_{opt}(N-i)$$

Also, for $1 \le j \le K$,

$$n_j^{(0)}_{opt}(N-i) \ge n_j(N-i) \frac{\sum_{k=1}^{j} n_k^{(0)}_{opt}(N-i)}{\sum_{k=1}^{j} n_k(N-i)}$$

$$<==> \quad \frac{1}{j} \ge \frac{n_j(N-i)}{\sum_{k=1}^{j} n_k(N-i)}$$

Since $n_1(N-i) \ge n_2(N-i) \ge \cdots \ge n_K(N-i)$ due to the assumed indexing of the fixed-rate service centers (as can easily be established using the MVA equations by induction on the population level [Au 81]), this last relation holds. From the definition of majorization, this establishes the theorem. ∎

In theorem 3.2, the preservation of bottleneck majorization is used to establish a relationship between level i and level $i+1$ mean queue lengths computed during the pessimistic computation of the PBH algorithm.

Theorem 3.2:

$(n_1^{(i+1)}_{pess}(N), \cdots, n_K^{(i+1)}_{pess}(N))$ is bottleneck majorized by $(n_1^{(i)}_{pess}(N), \cdots, n_K^{(i)}_{pess}(N))$ for $N-1 \ge i \ge 0$.

Proof:

From lemma 3.4, it need only be shown that $(n_1^{(1)}_{pess}(N-i), \cdots, n_K^{(1)}_{pess}(N-i))$ is bottleneck majorized by $(n_1^{(0)}_{pess}(N-i), \cdots, n_K^{(0)}_{pess}(N-i))$.

From equations (3-4) to (3-6), and equation (3-8),

$$n_k^{(1)}_{pess}(N-i) = \frac{L_k}{Z+1+\frac{b(N-i-1)}{Z+b(N-i-1)}(N-i-1)L_b}(N-i) \qquad k \ne b \qquad (3\text{-}9)$$

and

$$n_b^{(1)}_{pess}(N-i) = \frac{L_b\left(1+\frac{b(N-i-1)}{Z+b(N-i-1)}(N-i-1)\right)}{Z+1+\frac{b(N-i-1)}{Z+b(N-i-1)}(N-i-1)L_b}(N-i)$$

Since $b(N-i) \geq 1+\frac{b(N-i-1)}{Z+b(N-i-1)}(N-i-1)L_b$ (as follows from $b(N-i) = 1+\frac{b(N-i)}{Z+b(N-i)}(N-i-1)L_b$ and $b(N-i) \geq b(N-i-1)$),

$$\sum_{k=1}^{K} n_k^{(0)}{}_{pess}(N-i) = \frac{b(N-i)}{Z+b(N-i)}(N-i)$$

$$\geq \frac{1+\frac{b(N-i-1)}{Z+b(N-i-1)}(N-i-1)L_b}{Z+1+\frac{b(N-i-1)}{Z+b(N-i-1)}(N-i-1)L_b}(N-i) = \sum_{k=1}^{K} n_k^{(1)}{}_{pess}(N-i)$$

and

$$n_b^{(0)}{}_{pess}(N-i) = \frac{b(N-i)}{Z+b(N-i)}(N-i)$$

$$\geq \frac{L_b(1+\frac{b(N-i-1)}{Z+b(N-i-1)}(N-i-1))}{Z+1+\frac{b(N-i-1)}{Z+b(N-i-1)}(N-i-1)L_b}(N-i) = n_b^{(1)}{}_{pess}(N-i)$$

Finally, from equation (3-9) and the fact that $n_k^{(0)}{}_{pess}(N-i) = 0$, $n_k^{(1)}{}_{pess}(N-i) \geq n_k^{(0)}{}_{pess}(N-i)$ for $k \neq b$. From the definition of bottleneck majorization, this establishes the theorem. ∎

Since bottleneck majorization is transitive, and $n_k^{(N)}{}_{pess}(N)$ is exact for all k, theorem 3.2 implies that $(n_1(N), \cdots, n_K(N))$ is bottleneck majorized by $(n_1^{(i)}{}_{pess}(N), \cdots, n_K^{(i)}{}_{pess}(N))$, $N \geq i \geq 0$, $N \geq 1$. Note that theorem 3.2 is stronger than theorem 3.1, since in addition to establishing a majorization result between level i performance measures of the PBH algorithm and the exact performance measures, theorem 3.2 implies a nesting in the PBH bound hierarchy. As i increases, the level i pessimistic mean queue length allocation monotonically becomes less and less pessimistic. In section 3.2, a modification of the PBH algorithm that yields the same nesting property for the optimistic mean queue length allocation will be proposed.

In the next subsection, theorems 3.1 and 3.2 will be used to establish relationships between the exact primary and secondary performance measures and the approximate performance measures yielded during the optimistic and

pessimistic computations of the PBH algorithm.

3.1.4. The Bounds Produced

Theorem 3.3 summarizes the primary and secondary performance measure bounds that can be derived from the PBH algorithm. Note that $U_{K+1}(N)$ is usually defined as $n_{K+1}(N)$; it is therefore omitted from the primary performance measures. Also, the bounds involving only quantities from the pessimistic computation are nested, in that they monotonically approach the exact values as the bound level increases. In the theorem, A_k denotes the number of service centers with loading greater than or equal to L_k ($|\{j : L_j \geq L_k, 1 \leq j \leq K\}|$). Note that $A_k \geq 1$ since at least service center k satisfies this condition. The subscripts *UB* and *LB* denote upper and lower bounds, respectively, as obtained from the PBH algorithm as detailed in the theorem (for example, $R_{LB}^{(i)}(N) = max[R_{opt}^{(i)}(N), a(N)]$).

Theorem 3.3:

For $N \geq i \geq 0$, $N \geq 1$,

[Primary Performance Measures]

$$R_{pess}^{(i)}(N) \geq R(N) \geq max[R_{opt}^{(i)}(N), a(N)] \tag{3-10}$$

$$\frac{N}{Z + R_{LB}^{(i)}(N)} \geq X(N) \geq X_{pess}^{(i)}(N) \tag{3-11}$$

$$X_{UB}^{(i)}(N) L_k \geq U_k(N) \geq U_k^{(i)}{}_{pess}(N) \qquad 1 \leq k \leq K \tag{3-12}$$

[Secondary Performance Measures]

$$min\left[\frac{U_b^{(i)}{}_{UB}(N)}{1 - U_b^{(i)}{}_{UB}(N)}, n_b^{(i)}{}_{pess}(N)\right] \geq n_b(N) \geq n_b^{(i)}{}_{opt}(N) \tag{3-13}$$

$$min\left[\frac{L_b}{1 - U_b^{(i)}{}_{UB}(N)}, R_b^{(i)}{}_{pess}(N)\right] \geq R_b(N) \geq R_b^{(i)}{}_{opt}(N) \tag{3-14}$$

$$X_{UB}^{(i)}(N) Z \geq n_{K+1}(N) \geq n_{K+1}^{(i)}{}_{pess}(N) \tag{3-15}$$

$$min\left[\frac{U_k^{(i)}{}_{UB}(N)}{1 - U_k^{(i)}{}_{UB}(N)}, n_k^{(i)}{}_{pess}(N) + \frac{1}{A_k - 1}(n_b^{(i)}{}_{pess}(N) - n_b^{(i)}{}_{opt}(N))\right]$$

$$\geq n_k(N) \geq n_k^{(i)}{}_{pess}(N) \qquad k \neq b, 1 \leq k \leq K \tag{3-16}$$

$$min\left[\frac{L_k}{1-U_k^{(i)}{}_{JB}(N)},\ R_k^{(i)}{}_{pess}(N) + \frac{L_k}{(A_k-1)L_b}(R_b^{(i)}{}_{pess}(N) - R_b^{(i)}{}_{opt}(N))\right]$$

$$\geq R_k(N) \geq R_k^{(i)}{}_{pess}(N) \qquad k \neq b, 1 \leq k \leq K \tag{3-17}$$

Proof:

The bound of $a(N)$ in relation (3-10) follows from the correctness of ABA. (This bound is needed here since the optimistic computation lacks the nesting property of the pessimistic computation.) The bounds on secondary performance measures derived from $U_k^{(i)}{}_{JB}(N)$ $(1 \leq k \leq K)$ follow from Little's equation and the fact that

$$R_k(N) = L_k(1+n_k(N-1)) \leq L_k(1+n_k(N))$$

(Graham noted previously that this relation can be used to construct mean queue length bounds [Graham 81].)

In the remainder of the proof, each relation will be considered individually, ignoring the $a(N)$ and $U_k^{(i)}{}_{JB}(N)$ bounds. For $i=0$, these relations follow trivially from lemma 3.1, the correctness of ABA, and the fact that $L_j \geq L_k$ implies $R_j(N) \geq R_k(N)$ and $n_j(N) \geq n_k(N)$. Also, they are identities for $i=N=1$. Therefore, only $N \geq 2$, $i \geq 1$ need be considered.

[Primary Performance Measures]

[Relation 3-10]

Relation (3-10) can be expanded using equation (3-4) and the MVA mean residence time equation as

$$\sum_{k=1}^{K} L_k(1+n_k^{(i-1)}{}_{pess}(N-1)) \geq \sum_{k=1}^{K} L_k(1+n_k(N-1)) \geq \sum_{k=1}^{K} L_k(1+n_k^{(i-1)}{}_{opt}(N-1))$$

Applying theorems 3.1 and 3.2, and noting that bottleneck majorization implies majorization, this relation holds from relation (3.2-6) in lemma 3.2. (Note that for the upper bound, nesting is also shown, as is the case for all bounds involving only quantities from the pessimistic computation.)

[Relation 3-11]

From Little's equation, $X(N) = \frac{N}{Z+R(N)}$; an upper or lower bound on $R(N)$ therefore yields a lower or upper bound, respectively, on $X(N)$.

[Relation 3-12]

From Little's equation, $U_k(N) = X(N)L_k$ for all fixed-rate service centers k. An upper or lower bound on $X(N)$ therefore yields an upper or lower bound, respectively, on $U_k(N)$.

[Secondary Performance Measures]

[Relation 3-13]

Applying theorem 3.2, the upper bound holds from the definition of bottleneck majorization. The lower bound holds from theorem 3.1 and relation (3.2-3) in lemma 3.2.

[Relation 3-14]

From relation (3-13),

$$n_b^{(i-1)}{}_{pess}(N-1) \geq n_b(N-1) \geq n_b^{(i-1)}{}_{opt}(N-1)$$

Adding one to each term in this relation, multiplying through by L_b, and utilizing equation (3-4) and the MVA mean residence time equation yields the desired result.

[Relation 3-15]

From Little's equation, $n_{K+1}(N) = X(N)Z$. An upper or lower bound on $X(N)$ therefore yields an upper or lower bound, respectively, on $n_{K+1}(N)$.

[Relation 3-16]

The lower bound holds from theorem 3.2 and the definition of bottleneck majorization. The proof of the upper bound requires the following relation:

$$n_j^{(i+1)}{}_{pess}(N) - n_k^{(i+1)}{}_{pess}(N) \geq n_j^{(i)}{}_{pess}(N) - n_k^{(i)}{}_{pess}(N) \qquad (3\text{-}18)$$

for all fixed-rate service centers k and j such that $L_j \geq L_k$, $j \neq b$, $k \neq b$ $(N-1 \geq i \geq 0)$.

The proof of relation (3-18) is by induction on i. For $i=0$, the relation holds from equation (3-9), and from the fact that $n_j^{(0)}{}_{pess}(N) = n_k^{(0)}{}_{pess}(N) = 0$. Assume that the relation holds for $i-1$, and consider the relation for i $(i \geq 1)$. Noting that $X_{pess}^{(i+1)}(N) \geq X_{pess}^{(i)}(N)$, relation (3-18) will be shown for i if it can be shown when the left-hand side is divided by $X_{pess}^{(i+1)}(N)$, and the right-hand side is divided by $X_{pess}^{(i)}(N)$. Utilizing Little's equation and equation (3-4), this yields

$$L_j(1+n_j^{(i)}{}_{pess}(N-1)) - L_k(1+n_k^{(i)}{}_{pess}(N-1)) \geq L_j(1+n_j^{(i-1)}{}_{pess}(N-1)) - L_k(1+n_k^{(i-1)}{}_{pess}(N-1))$$

Since $n_k^{(i)}{}_{pess}(N-1) \geq n_k^{(i-1)}{}_{pess}(N-1)$, and $L_j \geq L_k$, this last relation will be shown if it can be shown when L_k is replaced by L_j. Simplification then yields relation (3-18) for $i-1$, $N-1$, which holds from the inductive hypothesis. This establishes relation (3-18) for i. By induction, relation (3-18) is established for all $i \geq 0$.

Relation (3-18) implies that

$$n_j(N) - n_k(N) \geq n_j^{(i)}{}_{pess}(N) - n_k^{(i)}{}_{pess}(N)$$

or

$$n_j(N) - n_j^{(i)}{}_{pess}(N) \geq n_k(N) - n_k^{(i)}{}_{pess}(N) \qquad (3\text{-}19)$$

Now, from relation (3-15), and the fact that $\sum_{k=1}^{K+1} n_k^{(i)}{}_{pess}(N) = \sum_{k=1}^{K+1} n_k(N)$,

$$\sum_{k=1}^{K} n_k^{(i)}{}_{pess}(N) \geq \sum_{k=1}^{K} n_k(N)$$

Relation (3-13) then yields

$$(n_b^{(i)}{}_{pess}(N) - n_b^{(i)}{}_{opt}(N)) + \sum_{k=1, k\neq b}^{K} n_k^{(i)}{}_{pess}(N) \geq \sum_{k=1, k\neq b}^{K} n_k(N)$$

The quantity $n_b^{(i)}{}_{pess}(N) - n_b^{(i)}{}_{opt}(N)$ is termed a *heap*. A heap represents a mean queue length quantity that is to be allocated to some set of service centers (in this case, the fixed-rate service centers, excluding the bottleneck service

center). Noting that $n_k^{(i)}{}_{pess}(N) \leq n_k(N)$ for $1 \leq k \leq K$, $k \neq b$, the heap represents a pessimistic value for the mean number of customers that have not been already accounted for by the $n_k^{(i)}{}_{pess}(N)$ values.

In the present context, the heap must be allocated so that the portion allotted to service center k is pessimistic. A separate heap allocation will be used for each service center, since an allocation that is pessimistic for one service center need not be pessimistic for some other service center. From relation (3-19), it is pessimistic for service center k to assume that all fixed-rate service centers with a higher or the same loading (excluding the bottleneck service center, since it is not under consideration here) receive an equal portion of the heap, and those with lesser loadings receive none of the heap. Relation (3-16) then follows directly.

[Relation 3-17]

From relation (3-16),

$$n_k^{(i-1)}{}_{pess}(N-1) + \frac{1}{A_k - 1}(n_b^{(i-1)}{}_{pess}(N-1) - n_b^{(i-1)}{}_{opt}(N-1)) \geq n_k(N-1) \geq n_k^{(i-1)}{}_{pess}(N-1)$$

Adding one to each queue length term in the relation, multiplying through by L_k, and utilizing equation (3-4) and the MVA mean residence time equation yields the desired result. ∎

It should be noted that, with the initial estimates of subsection 3.1.2, the level 1 PBH bounds on the primary performance measures correspond exactly to those yielded by the Balanced Job Bound algorithm, if the queueing network contains no delay service centers.

3.1.5. An Example

Table 3.2 gives the level 5 PBH bounds for the example queueing network whose loadings are given in table 3.1, for a population of 50 customers. Note

that the exact solution corresponds to bound level 50. Table 3.3 shows system throughput and mean system residence time bounds of various levels for the same population, illustrating the speed with which the PBH bounds approach the exact solution as the bound level increases. In figure 3.1, PBH bounds on system throughput and mean system residence time are shown over a range of workload intensities.

Fixed-rate Service Center Loadings							
1	2	3	4	5	6	7	8
.25	.20	.17	.16	.08	.06	.05	.03

Delay Service Center Loading: 15

An Example Queueing Network

Table 3.1

3.2. An Algorithm Modification

One disadvantage of the PBH algorithm as presented in section 3.1 is the lack of guaranteed nesting in the optimistic mean queue length allocations. (A vector of level i mean queue lengths may not be majorized by the level $i+1$ mean queue length vector for the same queueing network and population level.) Even so, the primary performance measure bounds produced by utilizing $\alpha(N)$ as well as quantities from the optimistic computation are empirically nested (although a proof of this has been elusive) In this section, the PBH algorithm is modified to guarantee nesting in the optimistic mean queue length allocations, and therefore in the bounds derived from the optimistic computation.

System Bounds					
System Throughput			Mean System Residence Time		
Upper	Exact	Lower	Upper	Exact	Lower
2.93	2.92	2.89	2.33	2.15	2.07

Fixed-Rate Service Center Bounds									
Service Center	Mean Residence Time			Mean Queue Length			Utilization		
	Upper	Exact	Lower	Upper	Exact	Lower	Upper	Exact	Lower
1	.93	.81	.74	2.74	2.35	2.16	.73	.73	.72
2	.48	.46	.43	1.41	1.33	1.23	.59	.58	.58
3	.34	.33	.31	.99	.95	.91	.50	.50	.49
4	.30	.29	.28	.88	.85	.82	.47	.47	.46
5	.10	.10	.10	.31	.30	.30	.23	.23	.23
6	.07	.07	.07	.21	.21	.21	.18	.17	.17
7	.06	.06	.06	.17	.17	.17	.15	.15	.14
8	.03	.03	.03	.10	.10	.09	.09	.09	.09

Delay Service Center Bounds		
Mean Queue Length		
Upper	Exact	Lower
43.94	43.74	43.28

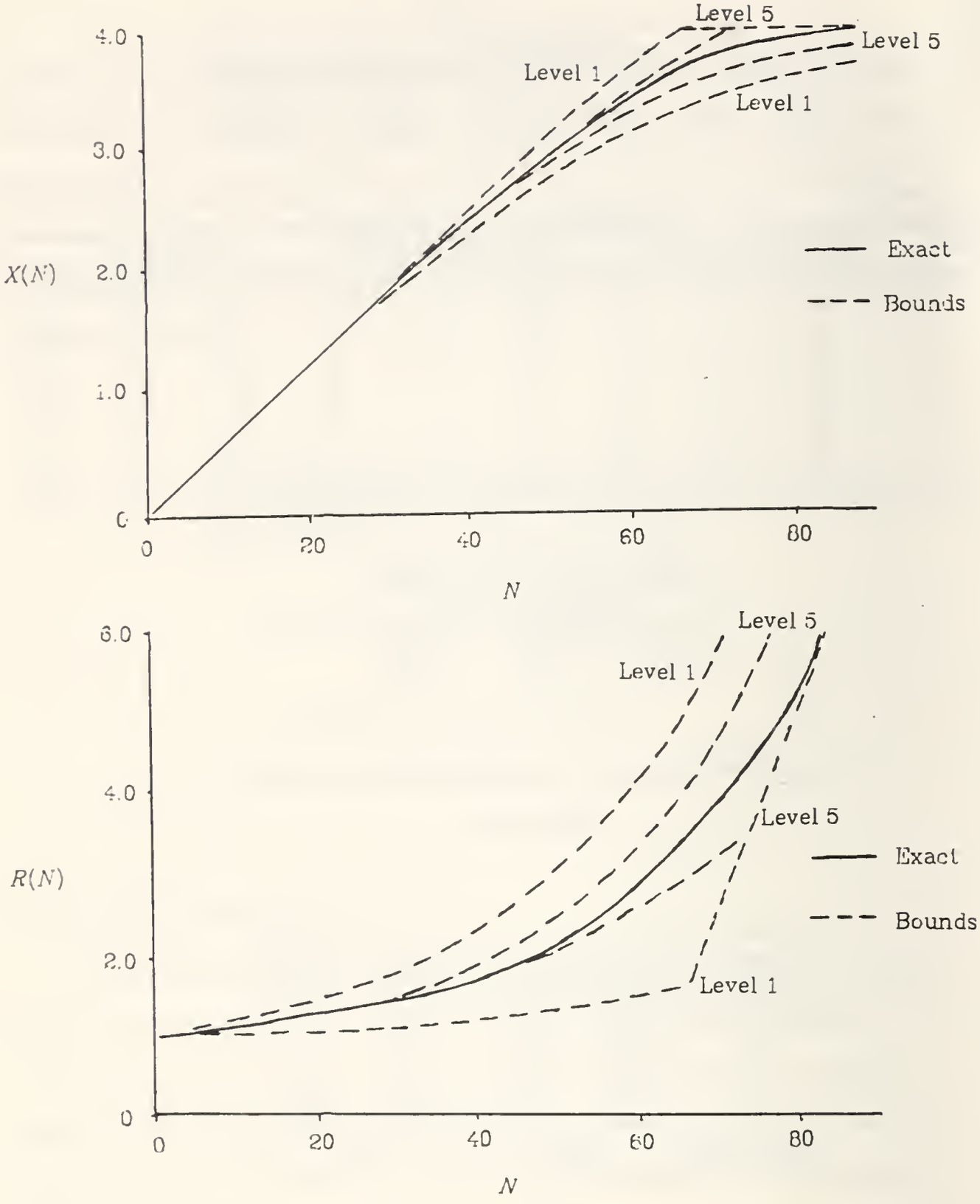
Level 5 PBH Bounds for an Example Queueing Network

Table 3.2

Bound Level	Bounds on $X(50)$		Bounds on $R(50)$	
	Upper	Lower	Upper	Lower
1	3.05	2.77	3.04	1.38
2	2.99	2.81	2.80	1.70
3	2.96	2.84	2.59	1.90
4	2.94	2.87	2.43	2.01
5	2.93	2.89	2.33	2.07
50 (exact)	2.92		2.15	

PEH Bounds of Various Levels for an Example Queueing Network

Table 3.3



PBH Bounds on System Throughput and Mean System Residence Time for an Example Queueing Network

Figure 3.1

This modification affects only the optimistic computation. While the optimistic computation of the basic PBH algorithm uses precisely the MVA equations, in the modified algorithm slightly different equations are used:

$$R_{k\,opt}^{(i)}(N) = L_k\left(1+n_{k\,opt}^{(i-1)}(N-1)\right) \qquad 1\le k\le K \tag{3-20}$$

$$n_{j\,opt}^{(i)}(N) = \left\lfloor \frac{R_{j\,opt}^{(i)}(N)}{\sum_{k=1}^{K} R_{k\,opt}^{(i)}(N)} \right\rfloor \frac{R_{opt}^{(i)}(N)}{Z+R_{opt}^{(i)}(N)}\, N \qquad 1\le j\le K \tag{3-21}$$

and

$$R_{opt}^{(i)}(N) = max\left[a(N)\,,\ \sum_{k=1}^{K} R_{k\,opt}^{(i)}(N)\right] \tag{3-22}$$

These equations are only marginally more expensive than those of the basic algorithm, and, in fact, have identical dominant terms in their time and space complexities (see section 3.4).

Nesting may be violated in the basic PBH algorithm due to the nature of the optimistic initial estimates. The optimistic initial estimation of the mean system residence time, and of the total mean number of customers at the fixed-rate service centers, is asymptotically exact as N increases since $a(N)$ is asymptotically exact. However, for large K the initial distribution of the fixed-rate service center customers (an equal allocation to each service center) is asymptotically very poor. This can induce poorer estimations, at the higher bound levels, of the mean system residence time and of the total mean number of customers at the fixed-rate service centers. Equations (3-20) to (3-22) compensate for this difficulty by utilizing $a(N)$ in the computation of these quantities. The equations reduce to the MVA equations for $N=i$ (since $max[R(N)\,,\,a(N)] = R(N)$), and, except for equation (3-22), for $Z=0$ as well. A proof that they preserve queue length majorization is given in lemma 3.5 of appendix 3.A. This result is utilized in theorem 3.4, which establishes nesting in the optimistic mean queue length allocations of the modified PBH algorithm.

Theorem 3.4:

With the modified PBH algorithm, $(n_{1\,opt}^{(i)}(N), \cdots, n_{K\,opt}^{(i)}(N))$ is majorized by $(n_{1\,opt}^{(i+1)}(N), \cdots, n_{K\,pess}^{(i+1)}(N))$ for $N-1 \geq i \geq 0$.

Proof:

From lemma 3.5, it need only be shown that $(n_{1\,opt}^{(0)}(N-i), \cdots, n_{K\,opt}^{(0)}(N-i))$ is majorized by $(n_{1\,opt}^{(1)}(N-i), \cdots, n_{K\,opt}^{(1)}(N-i))$. Consider then an indexing of the fixed-rate service centers such that $L_k \geq L_{k+1}$ for $1 \leq k \leq K-1$.

From equations (3-7) and (3-20) to (3-22), for $1 \leq k \leq K$

$$n_{k\,opt}^{(1)}(N-i) = L_k \frac{max\left[a(N-i),\ 1+\frac{1}{K}\frac{a(N-i-1)}{Z+a(N-i-1)}(N-i-1)\right]}{Z+max\left[a(N-i),\ 1+\frac{1}{K}\frac{a(N-i-1)}{Z+a(N-i-1)}(N-i-1)\right]}(N-i) \qquad (3\text{-}23)$$

Therefore,

$$\sum_{k=1}^{K} n_{k\,opt}^{(1)}(N-i) = \frac{max\left[a(N-i),\ 1+\frac{1}{K}\frac{a(N-i-1)}{Z+a(N-i-1)}(N-i-1)\right]}{Z+max\left[a(N-i),\ 1+\frac{1}{K}\frac{a(N-i-1)}{Z+a(N-i-1)}(N-i-1)\right]}(N-i)$$

$$\geq \frac{a(N-i)}{Z+a(N-i)}(N-i) = \sum_{k=1}^{K} n_{k\,opt}^{(0)}(N-i)$$

Also, for $1 \leq j \leq K$,

$$n_{j\,opt}^{(0)}(N-i) \geq n_{j\,opt}^{(1)}(N-i)\frac{\sum_{k=1}^{j} n_{k\,opt}^{(0)}(N-i)}{\sum_{k=1}^{j} n_{k\,opt}^{(1)}(N-i)}$$

reduces to, after using equations (3-7) and (3-23) and canceling common factors,

$$\frac{1}{j} \geq \frac{L_j}{\sum_{k=1}^{j} L_k}$$

which holds due to the assumed service center indexing. From the definition of majorization, this establishes the theorem. ∎

Theorem 3.4 also establishes that, with the modified PBH algorithm, $(n_1^{(i)}{}_{opt}(N), \cdots, n_K^{(i)}{}_{opt}(N))$ is majorized by $(n_1(N), \cdots, n_K(N))$ $(N \geq i \geq 0, N \geq 1)$, as was established in theorem 3.1 for the basic algorithm. This follows since $n_k^{(N)}{}_{opt}(N)$ is exact, and since majorization is transitive when all of the quantities involved are non-negative. Theorem 3.3 therefore holds unchanged for the modified algorithm (although it is redundant to explicitly consider $\alpha(N)$ in relation (3-10)).

It is easy to show that the modified optimistic computation produces level i performance measures that are at least as accurate (in the sense of being optimistic to the same or a lesser degree) as those of the basic optimistic computation. Empirically, however, the difference in primary performance measure values (if present at all) is significant only for queueing networks with many fixed-rate service centers, as long as the bound $\alpha(N)$ is used as in theorem 3.3. Secondary performance measure values differ more frequently and to a greater extent. Therefore, the modified optimistic computation is recommended only when secondary performance measure bounds are required.

In table 3.4, a comparison is made among the exact system and fixed-rate service center performance measures for the queueing network of table 3.1 and a population of 70 customers, the level 3 performance measures yielded by the modified optimistic computation, and the level 3 performance measures yielded by the basic optimistic computation (utilizing $\alpha(N)$ as in theorem 3.3). This choice of workload intensity and PBH performance measure level was one of the few for which the primary performance measures of the basic and the modified optimistic computation differed. Table 3.5 presents the same comparison under the assumption that an infinite number of fixed-rate service centers with zero loading have been added to the network. This modification causes the initial optimistic mean queue length allocation to be very poor, an effect which

propagates to higher levels in the basic optimistic computation.

3.3. Symbolic Formulation

3.3.1. Algorithm Structure

The symbolic formulation of the PBH algorithm utilizes symbolic MVA as its basis, rather than MVA. This modifies the computational method used; it does not change the performance measure values that are obtained.

For queueing networks of the type being treated, symbolic MVA computes $W(n,p)$ quantities, where $W(n,p)$ is defined by

$$W(n,p) = \sum_{k=1}^{K} L_k^p E_k(n-p) = \sum_{k=1}^{K} L_k^{p-1} R_k(n-p+1) \tag{3-24}$$

From equation (2-3),

$$W(n,p) = L^{(p)} + \frac{W(n,p+1)}{Z + W(n-p,1)}(n-p) \tag{3-25}$$

where $L^{(p)} = \sum_{k=1}^{K} L_k^p$. Symbolic MVA uses equation (3-25) to compute the mean system residence time $R(N) = W(N,1)$, starting from $W(n,p)$ values such that $n=p$ $(1 \le p \le N)$. Other primary performance measures (as well as the mean queue length for a delay service center) can be obtained using Little's equation. The mean residence time at each fixed-rate service center k of interest (and therefore, from Little's equation, the mean queue length) can be computed using (from equation (2-4))

$$W_k(n,p) = L_k^p + \frac{W_k(n,p+1)}{Z + W(n-p,1)}(n-p) \tag{3-26}$$

where

$$W_k(n,p) = L_k^p E_k(n-p) = L_k^{p-1} R_k(n-p+1)$$

The symbolic PBH algorithm computes $W^{(j)}(n,p)$ quantities, where

$$W^{(j)}(n,p) = \sum_{k=1}^{K} L_k^{p-1} R_k^{(j)}(n-p+1) \tag{3-27}$$

The symbolic formulation of the basic PBH algorithm utilizes equation (3-25)

System Performance Measures (Level 3)					
System Throughput			Mean System Residence Time		
Basic	Mod.	Exact	Basic	Mod.	Exact
3.96	3.95	3.72	2.69	2.74	3.84

Fixed-Rate Service Center Performance Measures (Level 3)								
Service Center	Mean Residence Time			Mean Queue Length			Utilization		
	Basic	Mod.	Exact	Basic	Mod.	Exact	Basic	Mod.	Exact
1	.97	1.00	2.00	3.84	3.93	7.45	.99	.99	.93
2	.60	.61	.72	2.38	2.42	2.67	.79	.79	.74
3	.43	.44	.45	1.72	1.74	1.66	.67	.67	.63
4	.39	.39	.38	1.53	1.55	1.43	.63	.63	.59
5	.12	.12	.11	.48	.48	.42	.32	.32	.30
6	.08	.08	.08	.32	.32	.29	.24	.24	.22
7	.06	.06	.06	.25	.25	.23	.20	.20	.19
8	.03	.03	.03	.14	.14	.13	.12	.12	.11

A Comparison Between the Basic and Modified PBH Optimistic Iteration

Table 3.4

System Performance Measures (Level 3)					
System Throughput			Mean System Residence Time		
Basic	Mod.	Exact	Basic	Mod.	Exact
4.00	3.95	3.72	2.50	2.74	3.84

Fixed-Rate Service Center Performance Measures (Level 3)								
Service Center	Mean Residence Time			Mean Queue Length			Utilization		
	Basic	Mod.	Exact	Basic	Mod.	Exact	Basic	Mod.	Exact
1	.78	1.00	2.00	3.17	3.93	7.45	1.00	.99	.93
2	.51	.61	.72	2.05	2.42	2.67	.80	.79	.74
3	.38	.44	.45	1.52	1.74	1.66	.68	.67	.63
4	.34	.39	.38	1.37	1.55	1.43	.64	.63	.59
5	.12	.12	.11	.47	.48	.42	.32	.32	.30
6	.08	.08	.08	.32	.32	.29	.24	.24	.22
7	.06	.06	.06	.25	.25	.23	.20	.20	.19
8	.03	.03	.03	.14	.14	.13	.12	.12	.11

A Comparison Between the Basic and Modified PBH Optimistic Iteration (Large K)

Table 3.5

unchanged:

$$W^{(j)}(n,p) = L^{(p)} + \frac{W^{(j-1)}(n,p+1)}{Z + W^{(j-1)}(n-p,1)}(n-p) \tag{3-28}$$

Both an optimistic and a pessimistic computation are required. Note that both require $L^{(p)}$ values; these are computed once at the beginning of the algorithm and stored for later use. In the symbolic formulation of the modified PBH algorithm, the pessimistic computation utilizes equation (3-28), while, for the optimistic computation, equations (3-27) and (3-20) to (3-22) yield

$$W_{opt}^{(j)}(n,p) = L^{(p)} + \left[\frac{W_{opt}^{(j-1)}(n,p+1)}{W_{opt}^{(j-1)}(n-p,1)}\right] \frac{R_{opt}^{(j-1)}(n-p)}{Z + R_{opt}^{(j-1)}(n-p)}(n-p) \tag{3-29}$$

where

$$R_{opt}^{(j)}(n) = max[a(n),\ W_{opt}^{(j)}(n,1)]$$

Both the symbolic basic and the symbolic modified PBH algorithm obtain level i primary performance measures for population level N ($1 \le i \le N-1$) from $W^{(i)}(N,1)$, which is computed recursively from $W^{(0)}(n,p)$ values such that $(N-1)-i = n-p$. From equation (3-27) and the optimistic and pessimistic initial estimates of subsection 3.1.2,

$$W_{opt}^{(0)}(n,p) = L^{(p-1)} \frac{a(n-p+1)}{K}$$

and

$$W_{pess}^{(0)}(n,p) = L\beta^{-1}b(n-p+1)$$

Secondary performance measures for each fixed-rate service center k of interest are obtained using

$$W_k^{(j)}(n,p) = L_k^p + \frac{W_k^{(j-1)}(n,p+1)}{Z + W^{(j-1)}(n-p,1)}(n-p)$$

in the basic algorithm and in the pessimistic computation of the modified algorithm, and

$$W_{k\ opt}^{(j)}(n,p) = L_k^p + \left[\frac{W_{k\ opt}^{(j-1)}(n,p+1)}{W_{opt}^{(j-1)}(n-p,1)}\right] \frac{R_{opt}^{(j-1)}(n-p)}{Z + R_{opt}^{(j-1)}(n-p)}(n-p)$$

in the modified algorithm optimistic computation. From

$$W_k^{(j)}(n,p) = L_k^{p-1} R_k^{(j)}(n-p+1)$$

and the optimistic and pessimistic initial estimates of subsection 3.1.2,

$$W_{k\,opt}^{(0)}(n,p) = L_k^{p-1}\frac{a(n-p+1)}{K} \qquad 1 \le k \le K$$

and

$$W_{k\,pess}^{(0)}(n,p) = \begin{cases} L_b^{p-1} b(n-p+1) & \text{for } k=b \\ 0 & \text{for } k \ne b, 1 \le k \le K \end{cases}$$

3.3.2. Advantages and Disadvantages

The symbolic formulation of the PBH algorithm has both advantages and disadvantages computationally in comparison to the non-symbolic formulation. The next section will examine the computational costs of both formulations in detail; here relative characteristics only are considered.

The primary disadvantage of the symbolic formulation is the additional computational expense required to obtain secondary performance measures. If tight secondary performance measure bounds are required for a significant fraction of the queueing network service centers, then the symbolic formulation is quite likely unsuitable. In particular, bounds on the performance measures of open classes can only be obtained if mean queue length bounds for the closed class are available (see section 3.6); therefore, the symbolic formulation may be unsuitable if bounds on open class performance measures are required.

If, on the other hand, only bounds on the primary performance measures of the closed class are required, the symbolic formulation is typically less costly than the non-symbolic formulation. For example, the results of the next section imply that level i primary performance measure bounds for a single population level can be obtained at less cost using the symbolic formulation if $i<3K$, a condition which would usually hold. In addition, a major advantage of the symbolic formulation is the partitioning of its computational requirements. Most of the cost is incurred in the computation of the $L^{(p)}$ values, which is

performed prior to the optimistic and pessimistic computations. This computation does not have to be repeated if it is decided to compute bounds at multiple bound hierarchy levels (increasing the efficiency of attempting to achieve some given accuracy constraint), or if bounds for multiple population levels are required (increasing the efficiency of, for example, computing PBH bounds for queueing network populations from 1 to N, which will prove useful in chapter six).

3.4. Computational Cost

The dominant terms in the time and space complexities of obtaining exact performance measures using the MVA algorithm (where the time complexity is determined by the number of arithmetic operations that must be performed) are $4KN$ and K, respectively. The PBH algorithm can substantially reduce this cost, as is shown by the dominant complexity terms given in table 3.6. Four usages of the PBH algorithm are considered, in which: (1) a single recursive computation is performed (using either the optimistic or pessimistic initial estimates), (2) both an optimistic and a pessimistic computation are performed, (3) both computations are performed d times to compute level $i-d+1$ to level i bounds on the performance measures of a given population level; or (4) both computations are performed d times to compute bounds at a given level on the performance measures of population levels $N-d+1$ to N. The dominant complexity terms shown are the same for both the basic and the modified PBH algorithms.

The most efficient method of computing level i PBH performance measures using the non-symbolic algorithm formulation is to compute the level 1 PBH performance measures from their analytic expressions, and then use them as initial estimates for the optimistic and pessimistic computations. The level 1 performance measures are used since they have essentially the same cost as

Non-symbolic Formulation				
	Single	Both	$i-d+1$ to i	$N-d+1$ to N
Time	$4Ki$	$8Ki$	$8K(i-\frac{d}{2})d$	$8Kid$
Space	K	K	K	K

Symbolic Formulation				
	Single	Both	$i-d+1$ to i	$N-d+1$ to N
Time	$i^2+2Ki+2ki$	$2i^2+2Ki+4ki$	$2d(i(i-d)+\frac{d^2}{3})+$ $2Ki+4k(i-\frac{d}{2})d$	$(2i^2+4ki)d+2Ki$
Space	$2i+ki$	$2i+ki$	$2i+ki$	$2i+ki$

k = number of fixed-rate service centers for which secondary performance measures are required

The Computational Cost of the PBH Algorithm

Table 3.6

the level 0 performance measures, due to simplifications arising from the fact that the optimistic and pessimistic level 0 mean queue length allocations to each fixed-rate service center k are independent of L_k. The quantities given in table 3.6 are independent of whether the level 0 or the level 1 performance measures are used as initial estimates (since they include dominant terms only), and follow directly from the time complexity ($4K$) and space complexity (K) of an application of the MVA equations.

The symbolic formulation results are based on a particular approach to evaluating the $W^{(j)}(n,p)$ quantities. (The method of computing the required $W_k^{(j)}(n,p)$ quantities is more straightforward and will not be elaborated on here.) This approach is based on the observation that $W^{(j)}(n,p)$ is defined in

terms of $W^{(j-1)}(m,q)$ quantities such that $n-p-1=m-q$. $W^{(i)}(N,1)$ is defined in terms of all quantities $W^{(i-1)}(m,q)$ such that $N-2=m-q$ $(1\leq q\leq 2)$, and, in general, all $W^{(j)}(n,p)$ quantities such that $(N-1)-(i-j)=n-p$ $(1\leq p\leq (i-j)+1)$ are collectively defined in terms of all $W^{(j-1)}(m,q)$ quantities such that $(N-1)-(i-j+1)=m-q$ $(1\leq q\leq (i-j)+2)$. Therefore, the required $W^{(j)}(n,p)$ quantities can be computed by first calculating all of the $W^{(0)}(n,p)$ values such that $(N-1)-i=n-p$. These values can then be used to calculate all of the $W^{(1)}(n,p)$ values such that $(N-1)-(i-1)=n-p$ (saving these in the storage used for the $W^{(0)}(n,p)$ quantities), and so on until $W^{(i)}(N,1)$ has been calculated. In all, the number of $W^{(j)}(n,p)$ values that must be calculated is of the order of $\frac{i^2}{2}$. Each of these requires 2 arithmetic operations (see equation 3-28, for example, and note that the number of $\frac{n-p}{Z+W^{(j-1)}(n-p,1)}$ expressions that must be evaluated is of the order of i and that therefore these computations can be disregarded), once the necessary $L^{(p)}$ quantities have been obtained (which require an additional order $2Ki$ operations).

3.5. Algorithm Accuracy

3.5.1. Accuracy Measures

There are a number of measures of bound tightness. The measure that will be utilized in this thesis is the magnitude of the maximum possible percent relative error in a bound-based point estimate, and will be termed the *maximal error*. Given a lower bound L, an upper bound U, and a point estimate A $(L\leq A\leq U)$, the maximal error is defined as $\max_{E\in[L,U]} \frac{|A-E|}{E} * 100\%$. The particular point estimate that is used when calculating the maximal error is that which minimizes this error measure, and can be shown to be the harmonic mean of the two bounds ($\frac{2UL}{U+L}$). The maximal error is then given by

$$\frac{U-L}{U+L} * 100\%$$

The maximal error is, by definition, greater than or equal to the actual relative error in the bound-based point estimate, and should be regarded as quite pessimistic as a measure of point estimate accuracy with respect to the measures that are used when studying non-bounding approximation techniques. However, the intended interpretation of the maximal error measure is as a measure of bound tightness, not as a measure of the actual point estimate accuracy. In practical applications, the exact solution, and therefore the actual point estimate accuracy, is unknown.

A queueing network bounding algorithm yields bounds on a number of different performance measures. In this thesis, bounding algorithm accuracy will be evaluated by applying the maximal error measure to the primary performance measure bounds, since the primary performance measures are usually those of most importance to the modeller. Since system throughput and the utilization of each service center are directly related through Little's equation, their maximal error measures are the same. Therefore, only system throughput and mean system residence time will be explicitly considered.

3.5.2. Analytic Results

The "worst case" accuracy of a bounding algorithm can be defined as the supremum of its maximal errors, over all possible queueing networks of the type being treated and all possible population levels. A worst case accuracy result will be presented here for the PBH algorithm (both the basic and the modified), for queueing networks with no delay service centers. This result depends on the following conjecture:

Conjecture:

The supremum of the level i ($i \geq 1$) PBH primary performance measure bound maximal errors, for queueing networks with no delay service centers, occurs in the limit as the number of fixed-rate service centers tends to infinity, $L_b \to 0$, $L_k / L_b \to 0$ for $k \neq b$, and $N \to 1/L_b$.

Discussion:

Intuitively, the conjecture states that the worst errors occur as: (1) the queueing network size grows without bound; (2) the loading of the bottleneck service center becomes insignificant with respect to the total loading (recall that the total loading is normalized to one), implying that the pessimistic initial mean queue length allocation becomes very poor; (3) each other fixed-rate service center loading becomes insignificant with respect to the bottleneck service center loading, implying that the queueing network becomes very "unbalanced"; and (4) the queueing network population grows without bound, while remaining at the intersection point of the ABA bounds NL_b and 1.

The conjecture has been proved for $i=1$ and $i=2$; appendix 3.B contains the proof for $i=2$.

Theorem 3.5:

If the conjecture given above holds, then the supremum of the maximal errors of the level i ($i \geq 1$) PBH primary performance measure bounds, for queueing networks with no delay service centers, is $1/(2i+1) * 100\%$.

Proof:

In appendix 3.B, it is proved that $R_{opt}^{(i)}(N) \to 1$ (note that $\alpha(N) = max[NL_b, 1] \to 1$ as well) and $R_{poss}^{(i)}(N) \to 1 + \frac{1}{i}$ for the limiting case of the previous conjecture (with both the modified and the basic algorithms). By applying the definition of

the maximal error, this yields the desired result for the mean system residence time bounds, and, with the use of Little's equation, for the system throughput bounds as well. ∎

For queueing networks with delay service centers, the supremum of the system throughput maximal errors (for a given Z) is probably different from that of the mean system residence time maximal errors. By noting that a delay service center can be simulated by a large number of fixed-rate service centers, each with a small loading, and that the PBH initial estimates are poorer when an actual delay service center is replaced by such a simulation, theorem 3.5 can be applied to bound these supremums. If the conjecture of this section holds, then the system throughput maximal error supremum for networks with delay service centers is less than or equal to $1/(2i+1) * 100\%$, while that for the mean system residence time is less than or equal to $(Z+1)/(2i+Z+1) * 100\%$.

3.5.3. Numerical Results

The accuracy of the PBH algorithm has also been studied by applying the basic algorithm to a number of test queueing networks. A set of 48 queueing networks was chosen, of which 24 had no delay service centers, 12 had a delay service center with a loading of 10, and 12 had a delay service center with a loading of 20. Within each group, one half had 20 fixed-rate service centers and the others had 60. The loading of each fixed-rate service center in each queueing network was derived by sampling from a uniform distribution between 0 and 1 using a pseudo-random number generator, and then raising the result to a power p. Values of p greater than 1 make the queueing network fixed-rate service center loading distribution more skewed, in the sense of having only a few heavily loaded service centers, while values of p less than 1 make the distribu-

tion more balanced. One third of the queueing networks were generated using a p value of 2, one third using a p value of .5, and the remainder using a p value of 1. Finally, the fixed-rate service center loadings of each queueing network were normalized to ensure that they summed to 1.

Each queueing network was analyzed with four population levels that depended on the queueing network. These were chosen to be the smallest populations that yielded exact bottleneck service center utilizations of at least 50%, 70%, 85% and 95%. Level 1, 3, 5 and 7 bounds were computed for each queueing network and each of its population levels. Over each of several chosen groupings of tests, the maximum and the average of the maximal errors for system throughput and mean system residence time were computed for each bound level.

Table 3.7 summarizes the results for the 192 combinations of queueing network and bottleneck service center utilization. Note that the maximum and the average of the maximal errors decrease with increasing bound level. Also, they are higher for the mean system residence time than for the system throughput.

In tables 3.8 to 3.11, only the level 3 bounds are considered. In each table, a parameter has been chosen (respectively, the exact bottleneck utilization, the delay service center loading, the fixed-rate service center loading distribution skewness, and the number of fixed-rate service centers), and the behavior of the maximal errors to parameter changes has been illustrated. As the tables show the maximum and the average of the maximal errors increase slightly as the skewness increases and as the number of fixed-rate service centers increases. The maximum and the average of the maximal errors in the mean system residence time increase as the delay service center loading increases substantially, while increasing and then decreasing for the system throughput.

Bound Level	Maximal Errors (%)			
	$R(N)$ Aver.	$R(N)$ Max.	$X(N)$ Aver.	$X(N)$ Max.
1	22.9	55.6	9.3	23.6
3	10.5	34.8	3.4	6.9
5	5.9	28.8	1.9	5.5
7	3.5	20.8	1.1	3.8

Maximal Errors for a Range of Bound Levels

Table 3.7

Bottleneck Utilization	Maximal Errors (Level 3 Bounds) (%)			
	$R(N)$ Aver.	$R(N)$ Max.	$X(N)$ Aver.	$X(N)$ Max.
.50	2.9	6.3	0.7	1.3
.70	8.7	18.3	2.4	3.6
.85	15.7	32.5	4.9	6.6
.95	14.4	34.8	5.8	6.9

Maximal Errors for a Range of Bottleneck Utilizations

Table 3.8

Delay Service Center Loading	Maximal Errors (Level 3 Bounds) (%)			
	$R(N)$ Aver.	$R(N)$ Max.	$X(N)$ Aver.	$X(N)$ Max.
0	3.5	6.6	3.5	6.6
10	14.8	26.3	3.7	6.9
20	19.9	34.8	3.0	6.5

Maximal Errors for a Range of Delay Service Center Loadings

Table 3.9

Loading Skewness (value of p)	Maximal Errors (Level 3 Bounds) (%)			
	$R(N)$ Aver.	$R(N)$ Max.	$X(N)$ Aver.	$X(N)$ Max.
0.5	9.4	28.6	3.3	6.5
1.0	10.7	32.4	3.5	6.6
2.0	11.2	34.8	3.5	6.9

Maximal Errors for a Range of p Values

Table 3.10

Number of Fixed-Rate Service Centers	Maximal Errors (Level 3 Bounds) (%)			
	$R(N)$ Aver.	$R(N)$ Max.	$X(N)$ Aver.	$X(N)$ Max.
20	10.1	33.4	3.3	6.6
60	10.8	34.8	3.6	6.9

Maximal Errors for two Different Queueing Network Sizes

Table 3.11

(Note that an increase in the delay service center loading, for a fixed bottleneck utilization, requires an increase in the customer population.) Finally, as the exact bottleneck utilization increases, the maximum and the average of the maximal errors increase, with one exception. However, this last effect is highly dependent on the bound level and on the range of utilizations considered. (As the queueing network population increases, while keeping other parameters fixed, the maximal errors tend to zero.)

3.6. Modifications to Allow Open Customer Classes

3.6.1. The Modified Algorithm Structure

The class of queueing networks that can be analyzed by the PBH algorithm can be extended to allow for any number of open customer classes in addition to the one closed class previously assumed.

In this context, the MVA solution algorithm obtains the primary and secondary performance measures of the closed class by solving a queueing network with a population consisting only of the closed class customers, and with the loading at each fixed-rate service center k inflated by $\frac{1}{1-\sum_{r \in O} L_{kr}\lambda_r}$ (assuming that the queueing network is stable, which requires that $\max_{1 \le k \le K} \sum_{r \in O} L_{kr}\lambda_r$ be less than 1) [Zahorjan & Wong 81]. PBH bounds for the closed class can be obtained by applying either the basic or the modified algorithm described previously to this closed single class queueing network. The only transformation of the outputs that is required is a deflation of the service center utilization bounds. (However, the inflated utilizations should be used when forming service center mean queue length and residence time bounds.)

MVA obtains the primary and secondary performance measures of each open class s from the following mean residence time equation [Zahorjan &

Wong 81]:

$$R_{ks}(N_c,\vec{\lambda}) = \frac{L_{ks}}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+n_{kc}(N_c,\vec{\lambda})) \qquad 1\le k\le K \tag{3-30}$$

The other primary and secondary performance measures can be obtained from Little's equation, and from

$$R_s(N_c,\vec{\lambda}) = \sum_{k=1}^{K} R_{ks}(N_c,\vec{\lambda}) \tag{3-31}$$

PBH bounds for each open class s can be obtained from the PBH closed class performance measure bounds (subscripted by *UB* and *LB*), as summarized in theorem 3.5. The class s system throughput, delay service center mean queue length, and service center utilizations have been disregarded here, as these follow directly from the queueing network parameter values.

Theorem 3.5:

For each open class s, $N_c \ge i \ge 0$, $N_c \ge 1$, and open class arrival rates that maintain stability,

[Primary Performance Measures]

$$\sum_{k=1}^{K} \frac{L_{ks}}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+n_{kc}^{(i)}{}_{UB}(N_c,\vec{\lambda})) \ge R_s(N_c,\vec{\lambda}) \ge$$

$$\sum_{k=1}^{K} \frac{L_{ks}}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+n_{kc}^{(i)}{}_{LB}(N_c,\vec{\lambda})) + L_{sc\ LB}^{h}\left[\frac{R_c^{(i)}{}_{LB}(N_c,\vec{\lambda})}{Z_c+R_c^{(i)}{}_{LB}(N_c,\vec{\lambda})}N_c - \sum_{k=1}^{K} n_{kc}^{(i)}{}_{LB}(N_c,\vec{\lambda})\right] \tag{3-32}$$

where, indexing the fixed-rate service centers so that $\frac{L_{kc}}{1-\sum_{r\in O} L_{kr}\lambda_r} \ge \frac{L_{k+1\,c}}{1-\sum_{r\in O} L_{k+1\,r}\lambda_r}$ for $1\le k\le K-1$,

$$L_{sc\ LB}^{h} = min\left[\frac{L_{1s}}{1-\sum_{r\in O} L_{1r}\lambda_r}, \min_{1\le j\le K-1} \frac{\sum_{k=2}^{j+1} \frac{L_{ks}}{1-\sum_{r\in O} L_{kr}\lambda_r}}{j}\right]$$

[Secondary Performance Measures]

$$\frac{L_{ks}\lambda_s}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+n_{kc}^{(i)}{}_{UB}(N_c,\vec{\lambda})) \geq n_{ks}(N_c,\vec{\lambda}) \geq \frac{L_{ks}\lambda_s}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+n_{kc}^{(i)}{}_{LB}(N_c,\vec{\lambda})) \qquad (3\text{-}33)$$

$$\frac{L_{ks}}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+n_{kc}^{(i)}{}_{UB}(N_c,\vec{\lambda})) \geq R_{ks}(N_c,\vec{\lambda}) \geq \frac{L_{ks}}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+n_{kc}^{(i)}{}_{LB}(N_c,\vec{\lambda})) \qquad (3\text{-}34)$$

Proof:

Except for the lower bound on the class s mean system residence time, the proofs of these relations follow directly from the correctness of the closed class bounds and from equations (3-30) and (3-31).

The lower bound on mean system residence time expression

$$\frac{R_c^{(i)}{}_{LB}(N_c,\vec{\lambda})}{Z_c+R_c^{(i)}{}_{LB}(N_c,\vec{\lambda})}N_c - \sum_{k=1}^{K} n_{kc}^{(i)}{}_{LB}(N_c,\vec{\lambda})$$

defines a heap. The heap represents an optimistic value for the mean number of customers of class c that have not already been accounted for by the $n_{kc}^{(i)}{}_{LB}(N_c,\vec{\lambda})$ values. $L_{sc\ LB}^{h}$ denotes an optimistic average (inflated) loading of class s customers when encountering customers from the class c heap. The fact that $L_{sc\ LB}^{h}$ is optimistic follows from the definition of $n_{ck}^{(i)}{}_{LB}(N_c,\vec{\lambda})$ given by theorem 3.3, and from relation (3-19). ∎

3.6.2. An Example

Table 3.13 gives the level 3 PBH bounds (using the modified algorithm) for the queueing network created by adding to the network of table 3.1 an open class with loadings as given in table 3.12, for a closed class population of 35 customers and an open class arrival rate of 2.0. Figure 3.2 shows PBH bounds on the open class mean system residence time, over a range of open class workload intensities.

Fixed-rate Service Center Loadings							
1	2	3	4	5	6	7	8
.08	.17	.25	.03	.20	.16	.06	.05

The Open Class Loadings of an Example Queueing Network

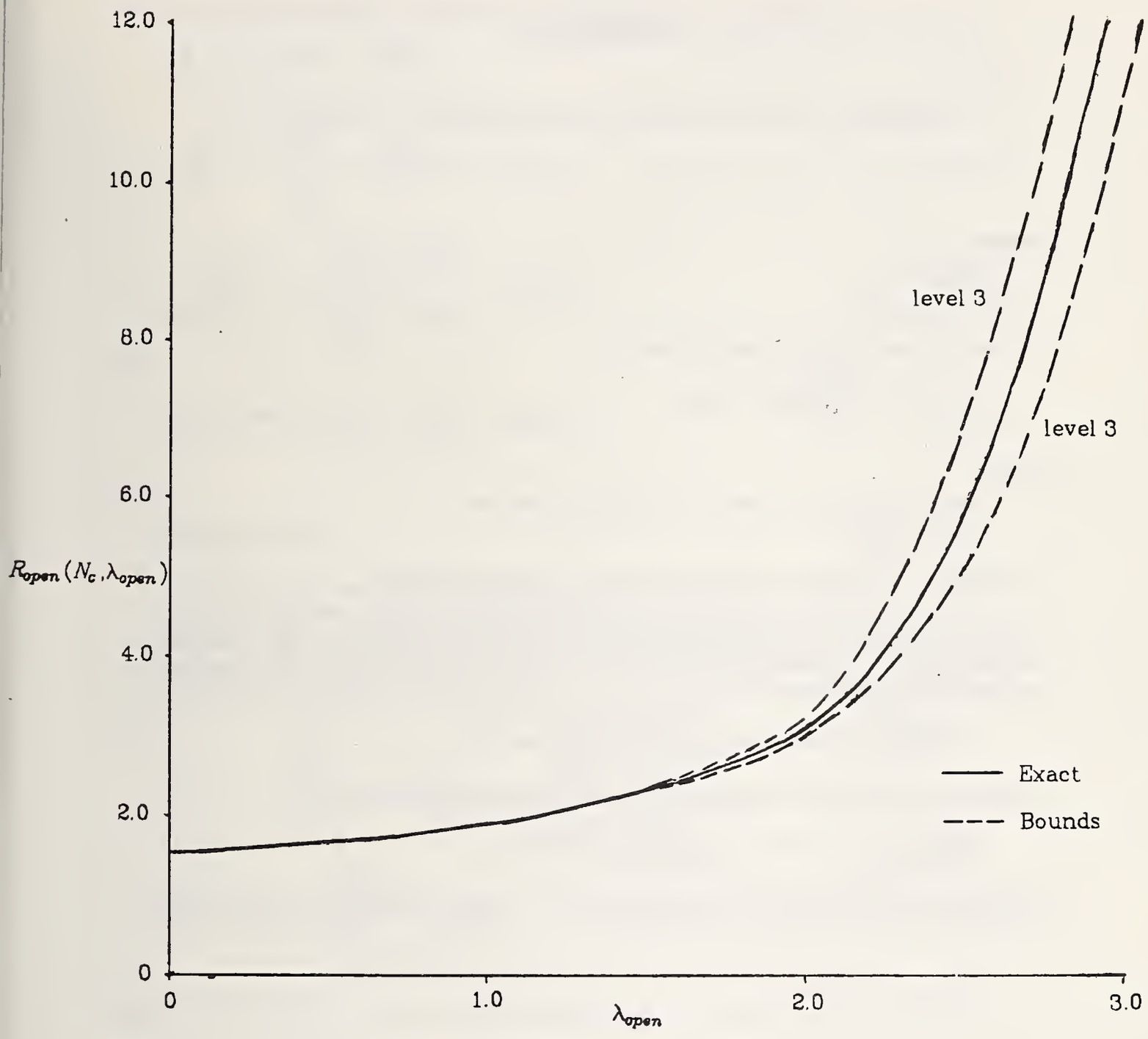
Table 3.12

System Bounds		
Mean System Residence Time		
Upper	Exact	Lower
3.34	3.13	2.95

Fixed-Rate Service Center Bounds						
Service Center	Mean Residence Time			Mean Queue Length		
	Upper	Exact	Lower	Upper	Exact	Lower
1	.23	.22	.20	.47	.44	.39
2	.65	.61	.54	1.29	1.21	1.08
3	1.54	1.38	1.26	3.07	2.77	2.53
4	.05	.05	.05	.10	.10	.09
5	.45	.45	.45	.91	.90	.89
6	.29	.28	.28	.57	.57	.57
7	.08	.08	.08	.15	.15	.15
8	.06	.06	.06	.12	.12	.12

Level 3 PBH Bounds on Open Class Performance Measures

Table 3.13



PBH Bounds on Open Class Mean System Residence Time

Figure 3.2

APPENDIX 3.A

This appendix states and proves lemmas of relevance to chapter three.

Lemma 3.1:

For all $N \geq 1$,

$$b(N) \geq R(N) \tag{3.1-1}$$

where $b(N)$ denotes the positive root of

$$b(N) = 1 + \frac{b(N)}{Z + b(N)}(N-1)L_b$$

Proof:

The lemma will be shown if it can be shown that

$$b(N) \geq R(n) \tag{3.1-2}$$

for $N \geq n \geq 1$. The proof of relation (3.1-2) is by induction on n.

For $n = 1$, relation (3.1-2) holds since $b(N) \geq 1 = R(1)$. Assume that the relation holds for $n-1$ ($N \geq n \geq 2$), and consider the relation for n. From the definition of $b(N)$, the MVA mean residence time equation, and Little's equation, relation (3.1-2) becomes

$$1 + \frac{b(N)}{Z + b(N)}(N-1)L_b \geq 1 + \sum_{k=1}^{K} \frac{R_k(n-1)}{Z + R(n-1)}(n-1)L_k$$

This last relation holds from the inductive hypothesis and from

$$\frac{R(n-1)}{Z + R(n-1)}(n-1)L_b \geq \sum_{k=1}^{K} \frac{R_k(n-1)}{Z + R(n-1)}(n-1)L_k$$

Relation (3.1-2) is therefore established for n, and, by induction, for all n such that $N \geq n \geq 1$. This establishes the lemma. ∎

Lemma 3.2:

Given quantities n_k $(1 \leq k \leq K)$, $n_k \geq n_{k+1} \geq 0$ $(1 \leq k \leq K-1)$, $n_1 > 0$, and quantities $\hat{n}_k$ $(1 \leq k \leq K)$, such that

$$\hat{n}_j \geq n_j \frac{\sum_{k=1}^{j} \hat{n}_k}{\sum_{k=1}^{j} n_k} \qquad 1 \leq j \leq K \tag{3.2-1}$$

and

$$\sum_{k=1}^{K} n_k \geq \sum_{k=1}^{K} \hat{n}_k \tag{3.2-2}$$

Then for any quantities L_k $(1 \le k \le K)$ such that $L_k \ge L_{k+1} \ge 0$ $(1 \le k \le K-1)$ and $L_1 > 0$,

$$\sum_{k=1}^{j} n_k \ge \sum_{k=1}^{j} \hat{n}_k \qquad 1 \le j \le K \tag{3.2-3}$$

$$\left(\sum_{k=1}^{j} L_k n_k\right) \frac{\sum_{k=1}^{j} \hat{n}_k}{\sum_{k=1}^{j} n_k} \ge \sum_{k=1}^{j} L_k \hat{n}_k \qquad 1 \le j \le K \tag{3.2-4}$$

$$1+\hat{n}_j \ge (1+n_j) \frac{\sum_{k=1}^{j} L_k (1+\hat{n}_k)}{\sum_{k=1}^{j} L_k (1+n_k)} \qquad 1 \le j \le K \tag{3.2-5}$$

and

$$\sum_{k=1}^{j} L_k (1+n_k) \ge \sum_{k=1}^{j} L_k (1+\hat{n}_k) \qquad 1 \le j \le K \tag{3.2-6}$$

Proof:

[Relation 3.2-3]

The proof is by downwards induction on j.

For $j=K$, $\sum_{k=1}^{K} n_k \ge \sum_{k=1}^{K} \hat{n}_k$ by assumption, establishing the induction basis.

Assume that relation (3.2-3) holds for $j+1$, $1 \le j \le K-1$, and consider the relation for j. Relation (3.2-1) applied for $j+1$ yields

$$\hat{n}_{j+1} \ge n_{j+1} \frac{\sum_{k=1}^{j+1} \hat{n}_k}{\sum_{k=1}^{j+1} n_k} \quad ==> \quad \sum_{k=1}^{j+1} \hat{n}_k - n_{j+1} \frac{\sum_{k=1}^{j+1} \hat{n}_k}{\sum_{k=1}^{j+1} n_k} \ge \sum_{k=1}^{j+1} \hat{n}_k - \hat{n}_{j+1}$$

$$==> \quad \left(\sum_{k=1}^{j} n_k\right) \frac{\sum_{k=1}^{j+1} \hat{n}_k}{\sum_{k=1}^{j+1} n_k} \ge \sum_{k=1}^{j} \hat{n}_k$$

Since $\sum_{k=1}^{j+1} n_k \ge \sum_{k=1}^{j+1} \hat{n}_k$ by the inductive hypothesis, it must be the case that $\sum_{k=1}^{j} n_k \ge \sum_{k=1}^{j} \hat{n}_k$; therefore, relation (3.2-3) is established for j. By induction, relation (3.2-3) is established for $1 \le j \le K+1$.

[Relation 3.2-4]

The proof is by induction on j.

For $j=1$, relation (3.2-4) trivially holds, establishing the induction basis.

Assume that relation (3.2-4) holds for j, $1 \le j < K$, and consider the relation for $j+1$. The relation will be shown if it can be shown for $\hat{n}_{j+1}$ minimized, since the assumed ordering of the L_k values yields

$$\sum_{k=1}^{j+1} L_k n_k - L_{j+1} \sum_{k=1}^{j+1} n_k \geq 0$$

which implies that

$$\hat{n}_{j+1}\left(\frac{\sum_{k=1}^{j+1} L_k n_k}{\sum_{k=1}^{j+1} n_k} - L_{j+1}\right)$$

is minimized for $\hat{n}_{j+1}$ minimized. By relation (3.2-1), the minimum $\hat{n}_{j+1}$ corresponds to

$$\hat{n}_{j+1} = n_{j+1} \frac{\sum_{k=1}^{j+1} \hat{n}_k}{\sum_{k=1}^{j+1} n_k}$$

which implies that

$$\frac{\sum_{k=1}^{j} \hat{n}_k}{\sum_{k=1}^{j} n_k} = \frac{\sum_{k=1}^{j+1} \hat{n}_k}{\sum_{k=1}^{j+1} n_k}$$

Using this last equality with the inductive hypothesis yields

$$\left(\sum_{k=1}^{j} L_k n_k\right) \frac{\sum_{k=1}^{j+1} \hat{n}_k}{\sum_{k=1}^{j+1} n_k} \geq \sum_{k=1}^{j} L_k \hat{n}_k$$

Since $L_{j+1} n_{j+1} \dfrac{\sum_{k=1}^{j+1} \hat{n}_k}{\sum_{k=1}^{j+1} n_k} = L_{j+1} \hat{n}_{j+1}$ by assumption, this last relation establishes relation (3.2-4) for $j+1$. By induction, relation (3.2-4) is established for $1 \leq j \leq K$.

[Relation 3.2-5]

Consider relation (3.2-5) for $1 \leq j \leq K$. From relations (3.2-1) and (3.2-4),

$$1+\hat{n}_j \geq (1+n_j) \frac{\sum_{k=1}^{j} L_k (1+\hat{n}_k)}{\sum_{k=1}^{j} L_k (1+n_k)}$$

holds if

$$1+n_j \frac{\sum_{k=1}^{j} \hat{n}_k}{\sum_{k=1}^{j} n_k} \geq (1+n_j) \frac{\sum_{k=1}^{j} L_k + \left(\sum_{k=1}^{j} L_k n_k\right) \dfrac{\sum_{k=1}^{j} \hat{n}_k}{\sum_{k=1}^{j} n_k}}{\sum_{k=1}^{j} L_k (1+n_k)}$$

Multiplying both sides by $(\sum_{k=1}^{j} n_k)\sum_{k=1}^{j} L_k(1+n_k)$ and simplifying produces

$$\sum_{k=1}^{j} L_k n_k (\sum_{k=1}^{j} n_k - \sum_{k=1}^{j} \hat{n}_k) \geq n_j \sum_{k=1}^{j} L_k (\sum_{k=1}^{j} n_k - \sum_{k=1}^{j} \hat{n}_k)$$

From relation (3.2-3), $\sum_{k=1}^{j} n_k - \sum_{k=1}^{j} \hat{n}_k \geq 0$; this fact applied to the preceding relation yields

$$\sum_{k=1}^{j} L_k n_k \geq n_j \sum_{k=1}^{j} L_k$$

Since $n_1 \geq n_2 \geq \cdots \geq n_K$ by assumption, this last relation holds, establishing relation (3.2-5) for $1 \leq j \leq K$.

[Relation 3.2-6]

Consider relation (3.2-6) for $1 \leq j \leq K$. From relation (3.2-4),

$$(\sum_{k=1}^{j} L_k n_k) \frac{\sum_{k=1}^{j} \hat{n}_k}{\sum_{k=1}^{j} n_k} \geq \sum_{k=1}^{j} L_k \hat{n}_k$$

Applying relation (3.2-3) yields

$$\sum_{k=1}^{j} L_k n_k \geq \sum_{k=1}^{j} L_k \hat{n}_k$$

which is equivalent to relation (3.2-6). This completes the proof of the lemma. ∎

Lemma 3.3:

Consider an indexing of the fixed-rate service centers such that $L_k \geq L_{k+1}$, $1 \leq k \leq K-1$. Suppose that two vectors of mean queue length values, $(n_1^{v(0)}(N), \cdots, n_K^{v(0)}(N))$ and $(n_1^{v'(0)}(N), \cdots, n_K^{v'(0)}(N))$ (where $n_k^{v(0)}(N) \geq n_{k+1}^{v(0)}(N) \geq 0$ for $1 \leq k \leq K-1$, $n_1^{v(0)}(N) > 0$ and $n_k^{v'(0)}(N) > -1$ for $1 \leq k \leq K$), are such that

$$n_j^{v'(0)}(N) \geq n_j^{v(0)}(N) \frac{\sum_{k=1}^{j} n_k^{v'(0)}(N)}{\sum_{k=1}^{j} n_k^{v(0)}(N)} \qquad 1 \leq j \leq K$$

and

$$\sum_{k=1}^{K} n_k^{v(0)}(N) \geq \sum_{k=1}^{K} n_k^{v'(0)}(N)$$

Then for all $i \geq 0$,

$$n_j^{v'(i)}(N+i) \geq n_j^{v(i)}(N+i) \frac{\sum_{k=1}^{j} n_k^{v'(i)}(N+i)}{\sum_{k=1}^{j} n_k^{v(i)}(N+i)} \qquad 1 \leq j \leq K \qquad (3.3\text{-}1)$$

and

$$\sum_{k=1}^{K} n_k^{v(i)}(N+i) \geq \sum_{k=1}^{K} n_k^{v'(i)}(N+i) \qquad (3.3\text{-}2)$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by i applications of the MVA equations starting from the values superscripted by $v(0)$ and $v'(0)$, respectively.

Proof:

The proof is by induction on i.

By assumption, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ ($i \geq 0$). From Little's equation and the MVA mean residence time equation, relation (3.3-2) reduces to

$$\sum_{k=1}^{K} \frac{L_k(1+n_k^{v(i)}(N+i))}{Z+\sum_{k=1}^{K} L_k(1+n_k^{v(i)}(N+i))}(N+i+1) \geq \sum_{k=1}^{K} \frac{L_k(1+n_k^{v'(i)}(N+i))}{Z+\sum_{k=1}^{K} L_k(1+n_k^{v'(i)}(N+i))}(N+i+1)$$

$$<==> \quad \sum_{k=1}^{K} L_k(1+n_k^{v(i)}(N+i)) \geq \sum_{k=1}^{K} L_k(1+n_k^{v'(i)}(N+i))$$

This last relation holds from lemma 3.2 applied to the inductive hypothesis, establishing relation (3.3-2) for $i+1$.

From Little's equation and the MVA mean residence time equation, for $1 \leq j \leq K$ relation (3.3-1) reduces to (after canceling common factors from both sides of the relation)

$$1+n_j^{v'(i)}(N+i) \geq (1+n_j^{v(i)}(N+i)) \frac{\sum_{k=1}^{j} L_k(1+n_k^{v'(i)}(N+i))}{\sum_{k=1}^{j} L_k(1+n_k^{v(i)}(N+i))}$$

This last relation holds from lemma 3.2 applied to the inductive hypothesis, establishing relation (3.3-1) and the lemma for $i+1$. By induction, the lemma is established for $i \geq 0$. ∎

Lemma 3.4:

Suppose that two vectors of non-negative mean queue length values, $(n_1^{v(0)}(N), \cdots, n_K^{v(0)}(N))$ and $(n_1^{v'(0)}(N), \cdots, n_K^{v'(0)}(N))$, are such that

$$n_b^{v'(0)}(N) \leq n_b^{v(0)}(N)$$

$$n_k^{v'(0)}(N) \geq n_k^{v(0)}(N) \qquad k \neq b$$

and

$$\sum_{k=1}^{K} n_k^{v(0)}(N) \geq \sum_{k=1}^{K} n_k^{v'(0)}(N)$$

Then for all $i \geq 0$,

$$n_b^{v'(i)}(N+i) \leq n_b^{v(i)}(N+i) \qquad (3.4\text{-}1)$$

$$n_k^{v'(i)}(N+i) \geq n_k^{v(i)}(N+i) \qquad k \neq b \qquad (3.4\text{-}2)$$

and

$$\sum_{k=1}^{K} n_k^{v(i)}(N+i) \geq \sum_{k=1}^{K} n_k^{v'(i)}(N+i) \qquad (3.4\text{-}3)$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by i applications of the MVA equations starting from the values superscripted by $v(0)$ and $v'(0)$, respectively.

Proof:

The proof is by induction on i.

By assumption, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ ($i \geq 0$). From Little's equation and the MVA mean residence time equation, relation (3.4-3) reduces to

$$\sum_{k=1}^{K} \frac{L_k(1+n_k^{v(i)}(N+i))}{Z+\sum_{k=1}^{K} L_k(1+n_k^{v(i)}(N+i))}(N+i+1) \geq \sum_{k=1}^{K} \frac{L_k(1+n_k^{v'(i)}(N+i))}{Z+\sum_{k=1}^{K} L_k(1+n_k^{v'(i)}(N+i))}(N+i+1)$$

This last relation holds if

$$\sum_{k=1}^{K} L_k(1+n_k^{v(i)}(N+i)) \geq \sum_{k=1}^{K} L_k(1+n_k^{v'(i)}(N+i))$$

$$<==> \quad L_b(n_b^{v(i)}(N+i)-n_b^{v'(i)}(N+i)) - \sum_{k \neq b} L_k(n_k^{v'(i)}(N+i)-n_k^{v(i)}(N+i)) \geq 0$$

From the inductive hypothesis on relations (3.4-1) and (3.4-2), each term in the above relation is non-negative. This implies that the relation will be established if it can be shown that

$$L_b(n_b^{v(i)}(N+i)-n_b^{v'(i)}(N+i)) - \sum_{k \neq b} L_b(n_k^{v'(i)}(N+i)-n_k^{v(i)}(N+i)) \geq 0$$

$$<==> \quad \sum_{k=1}^{K} n_k^{v(i)}(N+i) \geq \sum_{k=1}^{K} n_k^{v'(i)}(N+i)$$

This last relation holds from the inductive hypothesis on relation (3.4-3), establishing relation (3.4-3) for $i+1$.

From relation (3.4-3), relation (3.4-1) or (3.4-2) could be violated for $i+1$ only if there is a service center $j \neq b$ such that

$$n_j^{v(i+1)}(N+i+1) \geq n_j^{v'(i+1)}(N+i+1)$$

From Little's equation and the MVA mean residence time equation, this relation reduces to

$$\frac{L_j(1+n_j^{v(i)}(N+i))}{Z+\sum_{k=1}^{K} L_k(1+n_k^{v(i)}(N+i))}(N+i+1) \geq \frac{L_j(1+n_j^{v'(i)}(N+i))}{Z+\sum_{k=1}^{K} L_k(1+n_k^{v'(i)}(N+i))}(N+i+1)$$

$$<==> \quad \frac{1+n_j^{v(i)}(N+i)}{1+n_j^{v'(i)}(N+i)} \geq \frac{Z+\sum_{k=1}^{K} L_k(1+n_k^{v(i)}(N+i))}{Z+\sum_{k=1}^{K} L_k(1+n_k^{v'(i)}(N+i))}$$

However, the right-hand side of this last relation has been proved to be greater than or equal to one, while the left-hand side is less than or equal to one by the inductive hypothesis. As this is a contradiction, relations (3.4-1) and (3.4-2) and the lemma are established for $i+1$. By induction, the lemma is established

for $i \geq 0$. ∎

Lemma 3.5:

Consider an indexing of the fixed-rate service centers such that $L_k \geq L_{k+1}$, $1 \leq k \leq K-1$. Suppose that two vectors of mean queue length values, $(n_1^{v(0)}(N), \cdots, n_K^{v(0)}(N))$ and $(n_1^{v'(0)}(N), \cdots, n_K^{v'(0)}(N))$ (where $n_k^{v(0)}(N) \geq n_{k+1}^{v(0)}(N) \geq 0$ for $1 \leq k \leq K-1$, $n_1^{v(0)}(N) > 0$ and $n_k^{v'(0)}(N) > -1$ for $1 \leq k \leq K$), are such that

$$n_j^{v'(0)}(N) \geq n_j^{v(0)}(N) \frac{\sum_{k=1}^{j} n_k^{v'(0)}(N)}{\sum_{k=1}^{j} n_k^{v(0)}(N)} \qquad 1 \leq j \leq K$$

and

$$\sum_{k=1}^{K} n_k^{v(0)}(N) \geq \sum_{k=1}^{K} n_k^{v'(0)}(N)$$

Then for all $i \geq 0$,

$$n_j^{v'(i)}(N+i) \geq n_j^{v(i)}(N+i) \frac{\sum_{k=1}^{j} n_k^{v'(i)}(N+i)}{\sum_{k=1}^{j} n_k^{v(i)}(N+i)} \qquad 1 \leq j \leq K \tag{3.5-1}$$

and

$$\sum_{k=1}^{K} n_k^{v(i)}(N+i) \geq \sum_{k=1}^{K} n_k^{v'(i)}(N+i) \tag{3.5-2}$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by i applications of the equations used in the optimistic computation of the modified PBH algorithm (equations (3-20) to (3-22)), starting from the values superscripted by $v(0)$ and $v'(0)$, respectively.

Proof:

By induction on i.

By assumption, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ ($i \geq 0$). From equations (3-20) to (3-22), relation (3.5-2) reduces to

$$\frac{max\left[a(N+i+1),\ \sum_{k=1}^{K} L_k(1+n_k^{v(i)}(N+i))\right]}{Z+max\left[a(N+i+1),\ \sum_{k=1}^{K} L_k(1+n_k^{v(i)}(N+i))\right]}(N+i+1)$$

$$\geq \frac{max\left[a(N+i+1),\ \sum_{k=1}^{K} L_k(1+n_k^{v'(i)}(N+i))\right]}{Z+max\left[a(N+i+1),\ \sum_{k=1}^{K} L_k(1+n_k^{v'(i)}(N+i))\right]}(N+i+1)$$

which is equivalent to

$$\sum_{k=1}^{K} L_k(1+n_k^{v(i)}(N+i)) \geq \sum_{k=1}^{K} L_k(1+n_k^{v'(i)}(N+i))$$

This last relation holds from lemma 3.2 applied to the inductive hypothesis, establishing relation (3.5-2) for $i+1$.

From equations (3-20) and (3-21), for $1 \leq j \leq K$ relation (3.5-1) reduces to (after canceling common factors from both sides of the relation)

$$1+n_j^{v'(i)}(N+i) \geq (1+n_j^{v(i)}(N+i)) \frac{\sum_{k=1}^{j} L_k(1+n_k^{v'(i)}(N+i))}{\sum_{k=1}^{j} L_k(1+n_k^{v(i)}(N+i))}$$

This last relation holds from lemma 3.2 applied to the inductive hypothesis, establishing relation (3.5-1) and the lemma for $i+1$. By induction, the lemma is established for $i \geq 0$. ∎

APPENDIX 3.B

This appendix contains proofs of two results relevant to section 3.5: the conjecture of that section for $i=2$, and the values of $R_{opt}^{(i)}(N)$ and $R_{pess}^{(i)}(N)$ $(i \geq 1)$ in the limiting case of the conjecture.

Consider first the proof of the conjecture for $i=2$. For queueing networks with no delay service centers,

$$R_{opt}^{(2)}(N) = 1+L^{(2)}(N-1)$$

$$R_{pess}^{(2)}(N) = 1+\frac{L^{(2)}+L_b^2(N-2)}{1+L_b(N-2)}(N-1)$$

and

$$a(N) = max[NL_b\ ,\ 1]$$

Since the maximal error in a pair of bounds L and U $(\frac{U-L}{U+L} * 100\%)$ is maximized for $\frac{U}{L}$ maximized, the expression to be maximized is

$$\frac{1+\frac{L^{(2)}+L_b^2(N-2)}{1+L_b(N-2)}(N-1)}{max[1+L^{(2)}(N-1)\ ,\ NL_b]} \qquad (3.B\text{-}1)$$

It can be verified that

$$\frac{1+\frac{L^{(2)}+L_b^2(N-2)}{1+L_b(N-2)}(N-1)}{NL_b}$$

is maximized for N minimized for all $N \geq \frac{1}{L_b}$, by verifying that this claim holds if it holds for $L^{(2)}$ minimized, substituting L_b^2 for $L^{(2)}$ (since $L^{(2)} = \sum_{k=1}^{K} L_k^2$, the minimum $L^{(2)}$ is L_b^2), and finally taking the derivative of the resulting expression with respect to N. This result implies that expression (3.B-1) is maximized

at some point N (possibly non-integral) such that $1+L^{(2)}(N-1) \geq NL_b$. Therefore, expression (3.B-1) is maximized when, under the constraint on N,

$$\frac{1+\frac{L^{(2)}+L_b^2(N-2)}{1+L_b(N-2)}(N-1)}{1+L^{(2)}(N-1)} \tag{3.B-2}$$

is maximized.

It can be shown that the derivative of expression (3.B-2) with respect to N is non-negative if it is non-negative when $L^{(2)}$ is maximized. Substitution of the maximum $L^{(2)}$ (L_b) into (3.B-2) makes this expression identically 1. Therefore, expression (3.B-2) is maximized for N maximized, which corresponds to $1+L^{(2)}(N-1) = NL_b$, or $N = \frac{1-L^{(2)}}{L_b-L^{(2)}}$ (only $L^{(2)} > L_b$ need be considered). Substitution into (3.B-2) yields, after simplification,

$$\frac{L_b(1+L^{(2)}-2L_b)+2(L_b-L^{(2)})}{(1-L^{(2)})(2L_b-L^{(2)})}$$

This expression can be shown to be less than or equal to 3/2, using the fact that $L_b > L^{(2)}$. This maximum is attained in the limit as $L^{(2)} \to L_b^2$ and $L_b \to 0$ (implying that $L_k / L_b \to 0$ for all $k \neq b$, and that $N \to \frac{1}{L_b}$). This establishes the conjecture of section 3.5 for $i=2$.

It will also be shown in this appendix that $R_{opt}^{(i)}(N) \to 1$ and $R_{pess}^{(i)}(N) \to 1+\frac{1}{i}$ ($i \geq 1$) as $\frac{L_k}{L_b} \to 0$ for all $k \neq b$, $L_b \to 0$, and $N \to \frac{1}{L_b}$.

Consider first the claim that $R_{opt}^{(i)}(N) \to 1$. It will be shown, by induction on i, that $R_{opt}^{(i)}(N) \to 1$, $R_{k\,opt}^{(i)}(N) \to L_k$ for $k \neq b$, and $R_{b\,opt}^{(i)}(N) \to iL_b$. Only the basic algorithm will be treated explicitly, since $R_{opt}^{(i)}(N) = \alpha(N)$ implies the same result for the modified algorithm.

From the assumed limiting values of L_k ($1 \leq k \leq K$), it follows that $KL_b \to \infty$, implying that

$$R_{k\,opt}^{(1)}(N) = L_k\left(1+\frac{(N-1)}{K}\right) \to L_k$$

This establishes the induction basis. Assume that the claim holds for i. The claim for $i+1$ then follows from

$$R_{b\,opt}^{(i+1)}(N) = L_b\left(1+\frac{R_{b\,opt}^{(i)}(N-1)}{R_{opt}^{(i)}(N-1)}(N-1)\right) \to L_b\left(1+\frac{iL_b}{1}\frac{1}{L_b}\right) = (i+1)L_b$$

$$R_{k\,opt}^{(i+1)}(N) = L_k\left(1+\frac{R_{k\,opt}^{(i)}(N-1)}{R_{opt}^{(i)}(N-1)}(N-1)\right) \to L_k\left(1+\frac{L_k}{1}\frac{1}{L_b}\right) \to L_k \qquad k \neq b$$

and

$$R_{opt}^{(i+1)}(N) = \sum_{k=1}^{K} R_{k\,opt}^{(i+1)}(N) \to 1$$

By induction, the claim is established for all $i \geq 1$.

Finally, it will be shown that $R_{pess}^{(i)}(N) \to 1+\frac{1}{i}$, $R_{k\,pess}^{(i)}(N) \to L_k$ for $k \neq b$, and $R_{b\,pess}^{(i)}(N) \to \frac{i}{i}$ ($i \geq 1$). The proof is by induction on i.

Consider the claim for $i=1$.

$$R_b^{(1)}{}_{pess}(N) = L_b(1+(N-1)) \to 1$$

For $k \neq b$, $R_k^{(1)}{}_{pess}(N) = L_k$. Finally,

$$R_{pess}^{(1)}(N) = \sum_{k=1}^{K} R_k^{(1)}{}_{pess}(N) \to 2$$

establishing the induction basis. Assume that the claim holds for i. The claim for $i+1$ then follows from

$$R_b^{(i+1)}{}_{pess}(N) = L_b\left(1+\frac{R_b^{(i)}{}_{pess}(N-1)}{R_{pess}^{(i)}(N-1)}(N-1)\right) \to L_b\left(1+\frac{1}{i(1+\frac{1}{i})}\frac{1}{L_b}\right) \to \frac{1}{i+1}$$

$$R_k^{(i+1)}{}_{pess}(N) = L_k\left(1+\frac{R_k^{(i)}{}_{pess}(N-1)}{R_{pess}^{(i)}(N-1)}(N-1)\right) \to L_k\left(1+\frac{L_k}{1+\frac{1}{i}}\frac{1}{L_b}\right) \to L_k \qquad k \neq b$$

and

$$R_{pess}^{(i+1)}(N) = \sum_{k=1}^{K} R_k^{(i+1)}{}_{pess}(N) \to 1+\frac{1}{i+1}$$

By induction, the claim is established for all $i \geq 1$.

CHAPTER 4

THE MULTIPLE CLOSED CLASS PERFORMANCE BOUND HIERARCHY ALGORITHM

The PBH algorithm requires substantial modifications before it can be applied to queueing networks with multiple closed classes. For queueing networks with only a single closed class, the initial PBH performance measure estimates have the property that their optimistic or pessimistic character is preserved by the MVA equations. Unfortunately, it has not been possible to find a generally applicable estimation procedure, and suitable definitions of "optimistic" and "pessimistic", for which this property holds when there are multiple closed classes. The proposed solution to this problem is to modify the multiple class MVA equations to create equations that do preserve the character (under an appropriate definition) of initial performance measure estimates.

These modifications are presented in section one. Section two presents the symbolic formulation of the algorithm that results when symbolic MVA, rather than MVA, is modified to provide the algorithm structure. In section three, the computational cost of the multiple closed class PBH algorithm is determined both for the symbolic and the non-symbolic formulation. The accuracy of the algorithm is treated in section four. Section five contains a description of an error-bounded approximation algorithm, for which a variation of the multiple closed class PBH algorithm provides error bounds. Finally, in section six the algorithm is extended to allow for load-dependent service centers. The lemmas that are required in the body of the chapter are proved in appendix 4.A.

4.1. Algorithm Definition

4.1.1. Algorithm Structure

As for queueing networks with only a single closed class, for queueing networks with multiple closed classes the major step in the MVA solution algorithm requires the solution of a queueing network with no open classes. For this reason, in much of the development of the multiple closed class PBH algorithm it can be assumed that all classes are closed; this assumption will be made except when stated otherwise.

For queueing networks with multiple closed classes, modifications must be made to the definitions of "optimistic" and "pessimistic" performance measure estimates. These modifications are necessitated by the fact that, with the single class definitions, initial performance measure estimates that are optimistic or pessimistic for one class need not be similarly optimistic or pessimistic for some other class.

Consider an allocation (distinguished by v' superscripts) of non-negative mean queue length values $[n_{ks}^{v'}(\vec{N})]_{1\le k\le K, s\in C}$ with an associated *heap* $n_s^{hv'}(\vec{N})$ for each $s\in C$, such that each $n_{ks}^{v'}(\vec{N})$ is less than or equal to $n_{ks}(\vec{N})$. Each heap is a queue length quantity that is to be allocated to the fixed-rate service centers, but that has not been accounted for by the $[n_{ks}^{v'}(\vec{N})]_{1\le k\le K, s\in C}$ values. For example, if class s does not visit a delay service center, $n_s^{hv'}(\vec{N}) = N_s - \sum_{k=1}^{K} n_{ks}^{v'}(\vec{N})$. This mean queue length allocation is *looser* than another mean queue length allocation $[n_{ks}^{v}(\vec{N})]_{1\le k\le K, s\in C}$ with an associated heap $n_s^{hv}(\vec{N})$ for each $s\in C$ (again, such that each $n_{ks}^{v}(\vec{N})$ is less than or equal to $n_{ks}(\vec{N})$), if $n_{ks}^{v'}(\vec{N}) \le n_{ks}^{v}(\vec{N})$ for each $s\in C$ and service center k $(1\le k\le K)$. It is *heap-optimistic* with respect to the other mean queue length allocation (but not necessarily with respect to the exact allocation) if it is looser, and if, for all $s\in C$,

$$n_s^{hv'}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v'}(\vec{N}) \leq n_s^{hv}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v}(\vec{N})$$

It is *heap-pessimistic* with respect to the other allocation if it is looser, and if, for all $s \in C$,

$$n_s^{hv'}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v'}(\vec{N}) \geq n_s^{hv}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v}(\vec{N})$$

Intuitively, a looser mean queue length allocation provides less information about the fixed-rate service center mean queue lengths. A heap-optimistic mean queue length allocation is relatively optimistic in its estimation of the mean queue length that has not been accounted for by its approximate allocation, while a heap-pessimistic allocation is relatively pessimistic in this estimation.

The multiple closed class PBH algorithm yields mean queue length allocations with two, rather than only one, associated mean queue length heaps for each class. Choosing the optimistic heap of each class, in conjunction with the individual service center mean queue length estimates, yields an allocation that is heap-optimistic with respect to the exact allocation, while the pessimistic heaps yield a heap-pessimistic allocation. Since both optimistic and pessimistic properties are provided, only one recursive computation is required to yield both upper and lower performance measure bounds.

The equations used in the non-symbolic formulation of the PBH algorithm are modifications of those used in MVA. For queueing networks of the type being considered, the MVA mean residence time equation for service center k $(1 \leq k \leq K)$ and class s $(s \in C)$ is

$$R_{ks}(\vec{N}) = L_{ks}\left(1 + \sum_{r \in C} n_{kr}(\vec{N} - \vec{e}_s)\right)$$

From Little's equation,

$$n_{ks}(\vec{N}) = \frac{R_{ks}(\vec{N})}{Z_s + R_s(\vec{N})} N_s$$

where, by definition,

$$R_s(\vec{N}) = \sum_{k=1}^{K} R_{ks}(\vec{N})$$

Starting from a population consisting of no customers, the MVA algorithm applies these equations until the population level of interest is attained.

Level i PBH bounds for population level $\vec{N}$ ($i \leq N$, where $N = \sum_{r \in C} N_r$) are computed by choosing appropriate estimates for the performance measures for all feasible populations with i fewer customers, and then applying modifications of the MVA equations until population level $\vec{N}$ is attained. If $i = N$, the computation is equivalent to that of MVA. For $i < N$, the computational cost is reduced, but approximate rather than exact performance measures are produced.

The computation of level i PBH quantities from level $i-1$ PBH quantities can be intuitively described as follows. First of all, the MVA mean residence time equation is applied to level $i-1$ lower bounds on service center mean queue lengths to yield level i lower bounds on service center mean residence times. Appropriate sums of the latter bounds yield mean system residence time lower bounds for all classes. However, level $i-1$ optimistic and pessimistic heaps (lower and upper bounds, respectively, on the total of the fixed-rate service center mean queue lengths not accounted for by the mean queue length lower bounds) are also available. These heaps are used to increase each mean system residence time lower bound to yield a level i upper bound, and a better level i lower bound. For example, an upper bound for class s can be obtained by assuming that the pessimistic heaps are due to mean queue length underestimates at a fixed-rate service center where class s has its largest loading, thus inflating the mean system residence time of class s by the largest amount. (In practice, the class r pessimistic heap is allocated to such a service center only if the center is visited by class r.) Finally, level i lower mean queue length

bounds, and level i optimistic and pessimistic heaps, are obtained by applying Little's equation to the level i residence time bounds.

When all classes visit a fixed-rate service center where class s has its largest loading, for all classes s, additional intuition is available regarding the computation of pessimistic performance measures. If each class has its largest loading at some *common* fixed-rate service center, the computation is equivalent to applying the unmodified MVA equations from initial mean queue length estimates that bottleneck majorize the exact values, as in the pessimistic computation of the single closed class PBH algorithm. Otherwise, the computation can be viewed as operating on a modified network in which a hypothetical fixed-rate service center has been added. Each class has a loading at this service center identical to the largest of its normal fixed-rate service center loadings. The computation is equivalent to applying to this modified network the unmodified MVA equations (from initial mean queue length estimates that bottleneck majorize the exact values), except that the mean system residence times do not include the service time at the hypothetical service center. (Assuming a first-come-first-served service discipline, the mean system residence times include only the time spent *waiting* for service.)

Superscripting level i PBH quantities by (i), the equations utilized by the multiple closed class PBH algorithm are

$$R_{ks}^{(i)}(\vec{N}) = L_{ks}\left(1+\sum_{r\in C} n_{kr}^{(i-1)}(\vec{N}-\vec{e}_s)\right) \tag{4-1}$$

$$R_{s\,opt}^{(i)}(\vec{N}) = max\left[a_s^{(i)}(\vec{N}),\ \sum_{k=1}^{K} R_{ks}^{(i)}(\vec{N})+\sum_{r\in C} L_{sr}^{h}{}_{LB}\, n_{r}^{h\,(i-1)}{}_{opt}(\vec{N}-\vec{e}_s)\right] \tag{4-2}$$

$$R_{s\,pess}^{(i)}(\vec{N}) = \sum_{k=1}^{K} R_{ks}^{(i)}(\vec{N})+\sum_{r\in C} L_{sr}^{h}{}_{UB}\, n_{r}^{h\,(i-1)}{}_{pess}(\vec{N}-\vec{e}_s) \tag{4-3}$$

$$n_{ks}^{(i)}(\vec{N}) = \frac{R_{ks}^{(i)}(\vec{N})}{Z_s+R_{s\,pess}^{(i)}(\vec{N})}N_s \tag{4-4}$$

$$n_{s\,opt}^{h\,(i)}(\vec{N}) = \frac{R_{s\,opt}^{(i)}(\vec{N})}{Z_s+R_{s\,opt}^{(i)}(\vec{N})}N_s-\sum_{k=1}^{K} n_{ks}^{(i)}(\vec{N}) \tag{4-5}$$

and

$$n_{s\ pess}^{h(i)}(\vec{N}) = \frac{R_{s\ pess}^{(i)}(\vec{N})}{Z_s + R_{s\ pess}^{(i)}(\vec{N})} N_s - \sum_{k=1}^{K} n_{ks}^{(i)}(\vec{N}) \qquad (4\text{-}6)$$

where $a_s^{(i)}(\vec{N})$ denotes a level i optimistic asymptotic bound, $L_{sr\ LB}^{h}$ and $L_{sr\ UB}^{h}$ are respectively lower and upper bounds on the average loading of class s customers when encountering customers from the class r heap, and the subscripts *opt* and *pess* denote the mean queue length optimistic and pessimistic heaps, respectively, and quantities derived from these heaps. It is proved in lemmas 4.1 and 4.3 of appendix 4.A that the optimistic and pessimistic properties of a mean queue length allocation and its associated heaps are preserved by these equations.

The level i optimistic asymptotic bound is derived by noting that the utilization of each fixed-rate service center must be less than or equal to one, and that the mean system residence time of each class s must be greater than or equal to one. Using only the first of these constraints results in

$$\max_k \left[\frac{N_s}{Z_s + a_s^{(i)}(\vec{N})} L_{ks} + \sum_{r \neq s} \frac{N_r}{Z_r + R_{r\ pess}^{(i)}(\vec{N})} L_{kr} \right] = 1$$

Solving for $a_s^{(i)}(\vec{N})$ and then adding the second constraint yields

$$a_s^{(i)}(\vec{N}) = \max\left[\max_k \left| \frac{N_s}{1 - \sum_{r \neq s} \frac{N_r}{Z_r + R_{r\ pess}^{(i)}(\vec{N})} L_{kr}} L_{ks} \right| - Z_s \ , \ 1 \right] \qquad (4\text{-}7)$$

$L_{sr\ UB}^{h}$ is chosen as the maximum, over the fixed-rate service centers visited by class r, of the class s loadings. Note that choosing $L_{sr\ LB}^{h}$ is equivalent to choosing an allocation of the class r heap, across the fixed-rate service centers, that is optimistic for class s. It is clearly optimistic to allocate the entire class r heap to a fixed-rate service center visited by class r at which class s has its smallest loading. This choice makes $L_{sr\ LB}^{h}$ the minimum, over the fixed-rate service centers visited by class r, of the class s loadings. For the

initial estimates of the next subsection, however, additional information is available. Suppose that, for some fixed-rate service centers k and j, $L_{kt} \leq L_{jt}$ for all classes t. Theorem 4.1 of the next subsection then implies that the initial estimates given in that subsection produce a greater underestimate of the mean queue length at service center j than at service center k. This constrains the space of feasible optimistic heap allocations by the relation $h_j \geq h_k$, where h_j and h_k denote the allocations from the optimistic heaps to service centers j and k, respectively.

The problem of finding the most optimistic possible (under the constraints) allocation for a class s is a linear programming problem, where $\sum_{k=1}^{K} L_{ks} h_k$ is to be minimized. It is easily shown that each h_k can be chosen as either zero or some common value h. For suppose not; then let $j \geq 2$ be the service center index such that $0 = h_1 = \cdots = h_{j-i-1} < h_{j-i} = \cdots = h_{j-1} < h_j$, where the service centers are ordered such that their h_k values are in increasing order, and $1 \leq i \leq j-1$. If $\frac{(\sum_{k=j-i}^{j-1} L_{ks})}{i}$ is less than or equal to $\frac{(\sum_{k=j}^{K} L_{ks})}{K-j+1}$, then an allocation at least as optimistic would increase $h_{j-i}, \cdots, h_{j-1}$ and decrease $h_j, \cdots, h_K$ (while keeping the total allocation constant), until $h_{j-1} = h_j$. Similarly, if $\frac{(\sum_{k=j-i}^{j-1} L_{ks})}{i}$ is greater than or equal to $\frac{(\sum_{k=j}^{K} L_{ks})}{K-j+1}$, then an allocation at least as optimistic would decrease $h_{j-i}, \cdots, h_{j-1}$ to zero and increase $h_j, \cdots, h_K$. Note that these allocation changes can be performed without violating any constraints, implying a contradiction to the assertion that the solution of the allocation problem was as given.

Let $H_s = \{k : h_k = h\}$, where the h_k values minimize (subject to the con-

straints) $\sum_{k=1}^{K} L_{ks} h_k$, and where h is the common nonzero value. Then $L^h_{sr\ LB}$ can be chosen as the maximum of $\frac{(\sum_{k \in H_s} L_{ks})}{|H_s|}$ and the minimum, over the fixed-rate service centers visited by class r, of the class s loadings. The sets H_s, $s \in C$, can be computed by a linear program solution algorithm, such as the simplex method, prior to choosing a bound level and initiating the main PBH computation.

4.1.2. The Initial Estimates

This subsection presents one choice of appropriate initial performance measure estimates.

The initial mean queue length allocation for class s is the loosest possible:

$$R^{(0)}_{ks}(\vec{N}) = n^{(0)}_{ks}(\vec{N}) = 0 \qquad 1 \le k \le K \tag{4-8}$$

The chosen pessimistic initial estimate of the class s mean system residence time uses an extension of the pessimistic bound $b(N)$ to multiple closed class queueing networks:

$$R^{(0)}_{s\ pess}(\vec{N}) = b_s(\vec{N})$$

where $b_s(\vec{N})$ is defined by

$$b_s(\vec{N}) = 1 + \sum_{r \in C} \frac{b_r(\vec{N})}{Z_r + b_r(\vec{N})} (N_r - \delta_{rs}) L^h_{sr\ UB}$$

If no class visits a delay service center, the $b_s(\vec{N})$ values for all $s \in C$ can be obtained directly. Otherwise, the defining equations can be solved by successive substitution (any number of iterations may be used, depending on accuracy requirements), using $b_s(\vec{N}) = N$ for all $s \in C$ as initializations. It is proved in lemma 4.4 that this yields pessimistic mean system residence time bounds. Using equation (4-6), the initial pessimistic heap for class s is then given by

$$n^{h(0)}_{s\ pess}(\vec{N}) = \frac{b_s(\vec{N})}{Z_s + b_s(\vec{N})} N_s \tag{4-9}$$

The optimistic initial estimate of the class s mean system residence time uses the level 0 asymptotic bound given by equation (4-7):

$$R_{s\,opt}^{(0)}(\vec{N}) = a_s^{(0)}(\vec{N})$$

Using equation (4-5), the initial optimistic heap for class s is then given by

$$n_{s\,opt}^{h(0)}(\vec{N}) = \frac{a_s^{(0)}(\vec{N})}{Z_s + a_s^{(0)}(\vec{N})} N_s \tag{4-10}$$

The following theorem uses the preservation of the optimistic and pessimistic properties of a mean queue length allocation and its associated heaps, as proved in lemmas 4.1 and 4.3, to establish a relationship between level i and level $i+1$ mean queue lengths computed by the PBH algorithm. The theorem also shows that, with the initial estimates given here, the underestimate of the mean queue length at service center j is greater than that at service center k if $L_{kt} \le L_{jt}$ for all classes t, implying that $L_{ST\,LB}^{h}$ can be computed using the linear programming method from the previous subsection.

Theorem 4.1:

For $N-1 \ge i \ge 0$, $s \in C$,

$$n_{ks}^{(i)}(\vec{N}) \le n_{ks}^{(i+1)}(\vec{N}) \qquad 1 \le k \le K \tag{4-11}$$

$$n_{s\,pess}^{h(i)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{(i)}(\vec{N}) \ge n_{s\,pess}^{h(i+1)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{(i+1)}(\vec{N}) \tag{4-12}$$

$$n_{s\,opt}^{h(i)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{(i)}(\vec{N}) \le n_{s\,opt}^{h(i+1)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{(i+1)}(\vec{N}) \tag{4-13}$$

and, if $L_{kt} \le L_{jt}$ for all $t \in C$ $(1 \le k \le K,\ 1 \le j \le K)$,

$$n_{ks}^{(i+1)}(\vec{N}) - n_{ks}^{(i)}(\vec{N}) \le n_{js}^{(i+1)}(\vec{N}) - n_{js}^{(i)}(\vec{N}) \tag{4-14}$$

Proof:

From lemmas 4.1, 4.2, and 4.3, it need only be shown that, for all populations $\vec{n}$ resulting from the removal of i customers,

$$n_{ks}^{(0)}(\vec{n}) \le n_{ks}^{(1)}(\vec{n}) \qquad 1 \le k \le K$$

$$n_{s\,pess}^{h(0)}(\vec{n}) + \sum_{k=1}^{K} n_{ks}^{(0)}(\vec{n}) \ge n_{s\,pess}^{h(1)}(\vec{n}) + \sum_{k=1}^{K} n_{ks}^{(1)}(\vec{n})$$

$$n_s^{h(0)}_{opt}(\vec{n}) + \sum_{k=1}^{K} n_{ks}^{(0)}(\vec{n}) \le n_s^{h(1)}_{opt}(\vec{n}) + \sum_{k=1}^{K} n_{ks}^{(1)}(\vec{n})$$

and, if $L_{kt} \le L_{jt}$ for all $t \in C$ $(1 \le k \le K,\ 1 \le j \le K)$,

$$n_{ks}^{(1)}(\vec{n}) - n_{ks}^{(0)}(\vec{n}) \le n_{js}^{(1)}(\vec{n}) - n_{js}^{(0)}(\vec{n})$$

These relations are established by substituting in the expressions for the level 0 performance measures as given by equations (4-8) to (4-10), and the level 1 performance measures as obtained by applying the PBH equations given in the previous subsection. ∎

Theorem 4.1 implies a nesting in the PBH bound hierarchy. As i increases, the mean queue length allocation becomes less loose, while the pessimistic and optimistic heaps become less pessimistic and optimistic, respectively. Since relations (4-11) to (4-14) are transitive, and since the level N performance measures are exact, theorem 4.1 also holds when the level $i+1$ performance measures in the theorem are replaced by the exact values. In the next subsection, this fact will be used to establish bounds on primary and secondary performance measures.

4.1.3. The Bounds Produced

Theorem 4.2 summarizes the primary and secondary performance measure bounds that can be derived from the PBH algorithm. As implied by theorem 4.1, all bounds are nested. In the theorem, A_k denotes the number of service centers at which the loading of each class t is greater than or equal to L_{kt}. Since at least service center k satisfies this condition, $A_k \ge 1$.

Theorem 4.2:

For $N \ge i \ge 0$, $N_s \ge 1$, $s \in C$,

[Primary Performance Measures]

$$R_{s\,pess}^{(i)}(\vec{N}) \geq R_s(\vec{N}) \geq R_{s\,opt}^{(i)}(\vec{N}) \tag{4-15}$$

$$X_{s\,opt}^{(i)}(\vec{N}) \geq X_s(\vec{N}) \geq X_{s\,pess}^{(i)}(\vec{N}) \tag{4-16}$$

$$U_{ks\,opt}^{(i)}(\vec{N}) \geq U_{ks}(\vec{N}) \geq U_{ks\,pess}^{(i)}(\vec{N}) \qquad 1 \leq k \leq K \tag{4-17}$$

[Secondary Performance Measures]

$$n_{(K+1)s\,opt}^{(i)}(\vec{N}) \geq n_{(K+1)s}(\vec{N}) \geq n_{(K+1)s\,pess}^{(i)}(\vec{N}) \tag{4-18}$$

$$n_{ks}^{(i)}(\vec{N}) + \frac{1}{A_k} n_{s\,pess}^{h(i)}(\vec{N}) \geq n_{ks}(\vec{N}) \geq n_{ks}^{(i)}(\vec{N}) \qquad 1 \leq k \leq K \tag{4-19}$$

and, for $i \geq 1$,

$$R_{ks}^{(i)}(\vec{N}) + \frac{L_{ks}}{A_k} \sum_{r \in C} n_{r\,pess}^{h(i-1)}(\vec{N} - \vec{e}_s) \geq R_{ks}(\vec{N}) \geq R_{ks}^{(i)}(\vec{N}) \qquad 1 \leq k \leq K \tag{4-20}$$

Proof:

For $i=0$, these relations follow trivially. Consider $i \geq 1$.

[Primary Performance Measures]

Applying equations (4-5) and (4-6), the lower and upper bounds in relation (4-15) are shown by relations (4-13) and (4-12), respectively, in theorem 4.1 (recalling that the relations in theorem 4.1 also hold when the level $i+1$ performance measures are replaced by the exact performance measures). Relations (4-16) and (4-17) then follow from Little's equation.

[Secondary Performance Measures]

Relation (4-18) follows from relation (4-16) and Little's equation. The lower bound in relation (4-19) follows from relation (4-11) in theorem 4.1. From relation (4-14) in theorem 4.1, it is pessimistic, with respect to the mean queue length of class s customers at service center k, to assume that each service center at which the loading of each class t is greater than or equal to L_{kt} receives an equal portion of the class s pessimistic heap, and that all other service centers receive none of this heap. In conjunction with relation (4-12), this

establishes the upper bound in relation (4-19). Applying equation (4-1), relation (4-20) reduces to

$$L_{ks}\left(1+\sum_{r\in C} n_{kr}^{(i-1)}(\vec{N}-\vec{e}_s)\right) + \frac{L_{ks}}{A_k}\sum_{r\in C} n_r^{h\,(i-1)}{}_{pess}(\vec{N}-\vec{e}_s) \geq$$
$$L_{ks}\left(1+\sum_{r\in C} n_{kr}(\vec{N}-\vec{e}_s)\right) \geq L_{ks}\left(1+\sum_{r\in C} n_{kr}^{(i-1)}(\vec{N}-\vec{e}_s)\right)$$

Subtracting and then dividing through by L_{ks}, and separating the terms corresponding to each class, results in relation (4-19) for $i-1$ and $\vec{N}-\vec{e}_s$. This establishes relation (4-20). ∎

Queueing networks with open customer classes are treated in a similar manner as in the single closed class PBH algorithm. The open classes are removed, and the loading of each closed class at each fixed-rate service center k is inflated by $\frac{1}{1-\sum_{r\in O} L_{kr}\lambda_r}$. Bounds on the performance measures of each closed class can be obtained by applying the PBH algorithm to the resulting closed queueing network, and then utilizing theorem 4.2. The only transformation of the outputs that is required is a deflation of the service center utilization bounds. Theorem 4.3 summarizes the bounds for each open class that can be obtained from the performance measure bounds (subscripted by *UB* and *LB*), and the pessimistic and optimistic heaps (superscripted by *pess* and *opt*, respectively) for the closed classes.

Theorem 4.3: For each open class s, and open class arrival rates that maintain stability,

[Primary Performance Measures]

$$\sum_{k=1}^{K} \frac{L_{ks}}{1-\sum_{r\in O} L_{kr}\lambda_r}\left(1+\sum_{r\in C} n_{kr}^{(i)}{}_{LB}(\vec{N},\vec{\lambda})\right) + \sum_{r\in C} L_{sr}^{h}{}_{UB} n_r^{h\,(i)}{}_{pess}(\vec{N},\vec{\lambda}) \geq R_s(\vec{N},\vec{\lambda}) \geq$$
$$\sum_{k=1}^{K} \frac{L_{ks}}{1-\sum_{r\in O} L_{kr}\lambda_r}\left(1+\sum_{r\in C} n_{kr}^{(i)}{}_{LB}(\vec{N},\vec{\lambda})\right) + \sum_{r\in C} L_{sr}^{h}{}_{LB} n_r^{h\,(i)}{}_{opt}(\vec{N},\vec{\lambda}) \qquad (4\text{-}21)$$

where $L^h_{sr\ LB}$ and $L^h_{sr\ UB}$ are defined exactly as for the closed classes (using the inflated loadings for both the open and the closed classes)

[Secondary Performance Measures]

$$\frac{L_{ks}\lambda_s}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+\sum_{r\in C} n^{(i)}_{kr\ UB}(\vec{N},\vec{\lambda})) \geq n_{ks}(\vec{N},\vec{\lambda}) \geq \frac{L_{ks}\lambda_s}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+\sum_{r\in C} n^{(i)}_{kr\ LB}(\vec{N},\vec{\lambda})) \quad (4\text{-}22)$$

$$\frac{L_{ks}}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+\sum_{r\in C} n^{(i)}_{kr\ UB}(\vec{N},\vec{\lambda})) \geq R_{ks}(\vec{N},\vec{\lambda}) \geq \frac{L_{ks}}{1-\sum_{r\in O} L_{kr}\lambda_r}(1+\sum_{r\in C} n^{(i)}_{kr\ LB}(\vec{N},\vec{\lambda})) \quad (4\text{-}23)$$

Proof:

These relations follow directly from the correctness of the closed class bounds, the definitions of $L^h_{sr\ UB}$ and $L^h_{sr\ LB}$, and the pessimistic and optimistic nature of the pessimistic and optimistic heaps, respectively. ∎

4.1.4. An Example

Table 4.1 gives the loadings of an example queueing network. The queueing network has three classes, which can be interpreted as representing a transaction workload (class T), a batch workload (class B), and an interactive terminal user workload (class I). To illustrate the differences in the loading magnitudes that may be present among these classes, the time unit chosen is the same for each class. The results of the analysis of this queueing network, however, are presented using the time unit choice for each class that yields fixed-rate service center loadings of the class that sum to unity. To obtain class s bounds in the original time units, the class s mean system residence time bounds must be multiplied by the sum of the class s fixed-rate service center loadings, while the class s system throughput bounds must be divided by this sum. (Utilization and mean queue length bounds require no adjustment.)

Table 4.2 gives level 5 PBH bounds for a class T arrival rate of .1, a population of 3 class B customers, and a population of 35 class I customers. Bounds

Fixed-rate Service Center Loadings								
Class	1	2	3	4	5	6	7	8
T	5.0	4.0	3.4	3.2	1.6	1.2	1.0	.6
B	100.0	50.0	130.0	30.0	170.0	190.0	250.0	160.0
I	8.0	4.0	2.0	10.0	2.4	6.8	5.6	1.2

Delay Service Center Loading of class *I*: 1200

An Example Queueing Network

Table 4.1

have been shown on the system throughputs of classes *B* and *I* (note that the system throughput of class *T* is by assumption equal to its arrival rate, which is a given quantity), the mean system residence time of each class, the utilizations by classes *B* and *I* and the total utilization at each fixed-rate service center, and the mean queue length of each class at each fixed-rate service center.

The bounds shown in table 4.2 were computed with $L^h_{sr\ LB}$ chosen as the minimum, over the fixed-rate service centers visited by class r, of the class s loadings. This choice was made primarily to provide consistency with the algorithm accuracy tests in section 4.4. (See that section for the motivation behind making this choice.) If $L^h_{sr\ LB}$ is chosen using the linear programming method, the mean system residence time lower bounds, and the bounds derived from them (utilization upper bounds, for example), are improved. In this example, the improvements are only marginal; for example, the lower bound on the class *I* mean system residence time improves from 2.73 to 2.78.

In figure 4.1, PBH bounds on the class *I* system throughput and mean system residence time are shown over a range of class *I* workload intensities, with

System Throughput Bounds					
Class B			Class I		
Upper	Exact	Lower	Upper	Exact	Lower
1.48	1.44	1.39	1.07	1.06	1.06

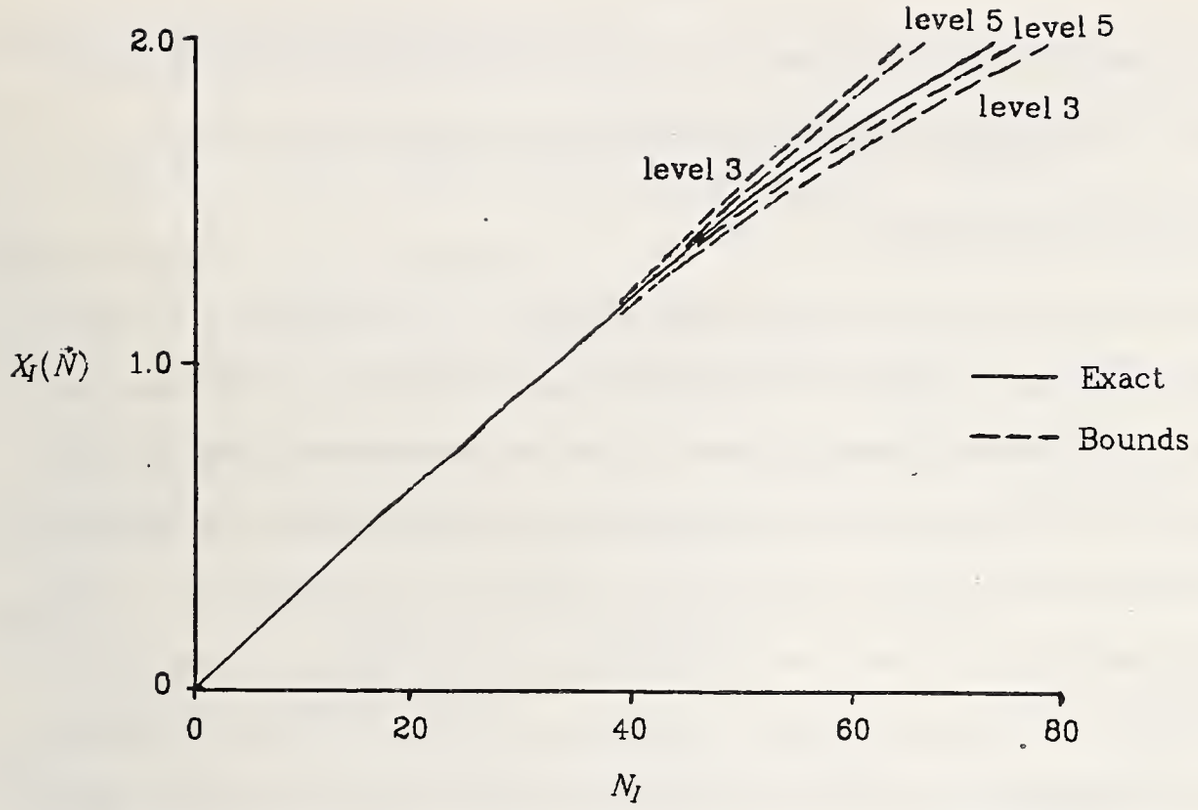
Mean System Residence Time Bounds								
Class T			Class B			Class I		
Upper	Exact	Lower	Upper	Exact	Lower	Upper	Exact	Lower
3.24	3.09	2.97	2.16	2.08	2.03	3.09	2.90	2.73

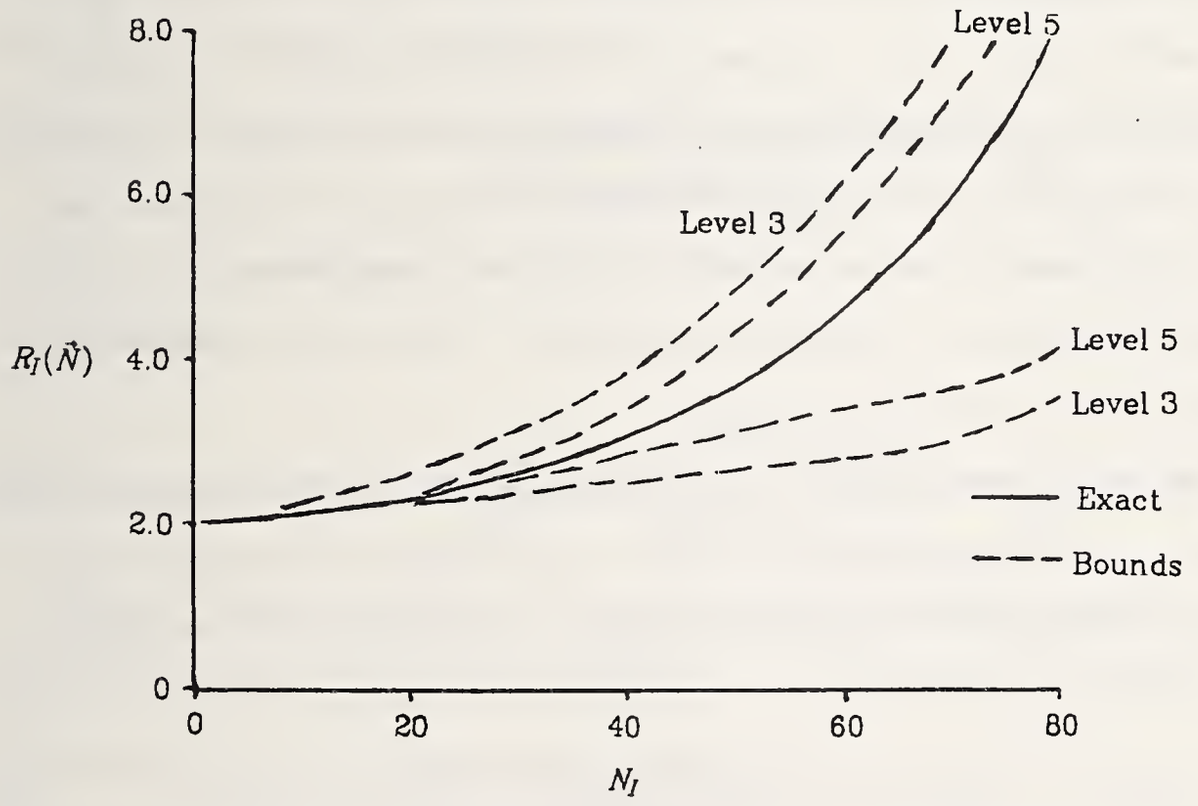
Fixed-Rate Service Center Utilization Bounds									
Service Center	Class B			Class I			Total over all classes		
	Upper	Exact	Lower	Upper	Exact	Lower	Upper	Exact	Lower
1	.14	.13	.13	.21	.21	.21	.85	.85	.84
2	.07	.07	.06	.11	.11	.11	.58	.57	.57
3	.18	.17	.17	.05	.05	.05	.57	.57	.56
4	.04	.04	.04	.27	.27	.26	.63	.63	62
5	.23	.23	.22	.06	.06	.06	.46	.45	.44
6	.26	.25	.24	.18	.18	.18	.56	.55	.54
7	.34	.33	.32	.15	.15	.15	.59	.58	.57
8	.22	.21	.21	03	.03	.03	.31	.31	.30

Fixed-Rate Service Center Mean Queue Length Bounds									
Service Center	Class T			Class B			Class I		
	Upper	Exact	Lower	Upper	Exact	Lower	Upper	Exact	Lower
1	3.20	2.82	2.62	.77	.65	.58	1.43	1.17	1.03
2	1.05	.93	.92	.20	.15	.14	.37	25	.24
3	.82	.76	.75	.38	.35	.34	.21	.12	.12
4	.96	.84	.83	.19	.10	.10	.86	.69	.67
5	.31	.28	.28	.41	.36	.35	.24	.11	.11
6	.32	.25	.25	.63	.48	.45	.75	.38	.36
7	.27	.22	.21	.77	.62	.59	.70	.32	.30
8	.09	.08	.08	.30	.28	.27	.11	.05	.04

Level 5 PBH Bounds for an Example Queueing Network

Table 4.2





PBH Bounds on Class *I* System Throughput and Mean System Residence Time

Figure 4.1

the workload intensities of classes T and B remaining fixed at an arrival rate of .1 and a population of 3, respectively. $L^{h}_{sr\ LB}$ was chosen as for table 4.2.

4.2. Symbolic Formulation

With the single closed class PBH algorithm, the performance measure values computed by the symbolic formulation are the same as those computed by the non-symbolic formulation. This is not always the case with the multiple closed class PBH algorithm. For computational efficiency reasons, in the symbolic formulation the $L^{h}_{sr\ UB}$ values must be chosen to be independent of r. The choice made for each class s, denoted by $L^{h}_{s\ UB}$, is the maximum of the class s fixed-rate service center loadings. (This value differs from $L^{h}_{sr\ UB}$, as chosen in the non-symbolic formulation, only if class r customers do not visit the fixed-rate service center with the largest class s loading.) Also, again for computational efficiency reasons, the symbolic formulation does not utilize the asymptotic bounds $a_s^{(i)}(\vec{N})$ until the final performance measures are computed at the end of the computation. In particular, $R_s^{(0)}{}_{opt}(\vec{N})$ is simply taken as 1. (This modification can also be applied to significantly reduce the computational expense of the non-symbolic formulation.)

The symbolic formulation of the multiple closed class PBH algorithm is based on a modification of symbolic MVA. For queueing networks of the type being treated, symbolic MVA computes $W(\vec{n},[s_1,\cdots,s_p])$ quantities using (equation (2-3))

$$W(\vec{n},[s_1,\cdots,s_p]) = L^{[s_1,\cdots,s_p]} + \sum_{r\in C} \frac{W(\vec{n},[s_1,\cdots,s_p,r])}{Z_r + W(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g},[r])}(n_r - \sum_{g=1}^{p}\delta_{rs_g})$$

where $W(\vec{n},[s_1,\cdots,s_p])$ is defined by

$$W(\vec{n},[s_1,\cdots,s_p]) = \sum_{k=1}^{K}(\prod_{g=1}^{p}L_{ks_g})F_k(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g}) = \sum_{k=1}^{K}(\prod_{g\neq l}L_{ks_g})R_{ks_l}(\vec{n}-\sum_{g\neq l}\vec{e}_{s_g}) \qquad 1\le l\le p$$

Secondary performance measures are computed for each fixed-rate service

center k of interest using (equation (2-4))

$$W_k(\vec{n},[s_1,\cdots,s_p]) = (\prod_{g=1}^{p} L_{ks_g}) + \sum_{r\in C} \frac{W_k(\vec{n},[s_1,\cdots,s_p,r])}{Z_r + W(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g},[r])}(n_r - \sum_{g=1}^{p}\delta_{rs_g})$$

where $W_k(\vec{n},[s_1,\cdots,s_p])$ is defined by

$$W_k(\vec{n},[s_1,\cdots,s_p]) = (\prod_{g=1}^{p} L_{ks_g})E_k(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g}) = (\prod_{g\neq l} L_{ks_g})R_{ks_l}(\vec{n}-\sum_{g\neq l}\vec{e}_{s_g}) \qquad 1\le l\le p$$

The symbolic formulation of the multiple closed class PBH algorithm computes $W^{(j)}_{pess}(\vec{n},[s_1,\ldots,s_p])$ quantities, where, for $j\ge 1$, $1\le l\le p$,

$$W^{(j)}_{pess}(\vec{n},[s_1,\cdots,s_p]) = \sum_{k=1}^{K}(\prod_{g\neq l} L_{ks_g})R^{(j)}_{ks_l}(\vec{n}-\sum_{g\neq l}\vec{e}_{s_g}) +$$

$$(\prod_{g=1}^{p} L^h_{s_g\ UB})\sum_{r\in C} n^{h\,(j-1)}_{r\ pess}(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g}) \qquad (4\text{-}24)$$

Using the defining equations of the non-symbolic formulation, it can be verified that $W^{(i)}_{pess}(\vec{N},[s])=R^{(i)}_{s\ pess}(\vec{N})$, and that, for $j>1$,

$$W^{(j)}_{pess}(\vec{n},[s_1,\cdots,s_p]) = L^{[s_1,\cdots,s_p]} + \sum_{r\in C}\frac{W^{(j-1)}_{pess}(\vec{n},[s_1,\cdots,s_p,r])}{Z_r + W^{(j-1)}_{pess}(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g},[r])}(n_r - \sum_{g=1}^{p}\delta_{rs_g}) \quad (4\text{-}25)$$

From the initial performance measure estimates of the previous section, and equation (4-24),

$$W^{(1)}_{pess}(\vec{n},[s_1,\cdots,s_p]) = L^{[s_1,\cdots,s_p]} + (\prod_{g=1}^{p} L^h_{s_g\ UB})\sum_{r\in C}\frac{b_r(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g})}{Z_r + b_r(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g})}(n_r - \sum_{g=1}^{p}\delta_{rs_g}) \quad (4\text{-}26)$$

$R^{(i)}_{s\ pess}(\vec{N})$ values are obtained from the $W^{(i)}_{pess}(\vec{N},[s])$ values, which are recursively computed starting from $W^{(1)}(\vec{n},[s_1,\cdots,s_p])$ quantities such that $\sum_{r\in C}(N_r-(n_r-\sum_{g=1}^{p}\delta_{rs_g})) = i$. From equations (4-3) to (4-6),

$$\sum_{k=1}^{K} R^{(i)}_{ks}(\vec{N}) = R^{(i)}_{s\ pess}(\vec{N}) - L^h_{s\ UB}\sum_{r\in C} n^{h\,(i-1)}_{r\ pess}(\vec{N}-\vec{e}_s)$$

$$\sum_{k=1}^{K} n^{(i)}_{ks}(\vec{N}) = \frac{R^{(i)}_{s\ pess}(\vec{N}) - L^h_{s\ UB}\sum_{r\in C} n^{h\,(i-1)}_{r\ pess}(\vec{N}-\vec{e}_s)}{Z_s + R^{(i)}_{s\ pess}(\vec{N})}N_s$$

and

$$n_{s\ pess}^{h\ (i)}(\vec{N}) = \frac{L_s^h{}_{UB} \sum_{r \in C} n_{r\ pess}^{h\ (i-1)}(\vec{N} - \vec{e}_s)}{Z_s + R_{s\ pess}^{(i)}(\vec{N})} N_s$$

These equations can be used, in conjunction with equation (4-2) (ignoring the $a_s^{(i)}(\vec{N})$ bounds until the final step) and equation (4-5), to recursively compute $R_{s\ opt}^{(i)}(\vec{N})$ for all $s \in C$ during the computation of the $R_{s\ pess}^{(i)}(\vec{N})$ values.

$R_{ks}^{(i)}(\vec{N})$, and therefore $n_{ks}^{(i)}(\vec{N})$, can be computed recursively, for all classes s and each fixed-rate service center k of interest, using, for $j > 1$,

$$W_k^{(j)}(\vec{n}, [s_1, \cdots, s_p]) = \prod_{g=1}^{p} L_{ks_g} + \sum_{r \in C} \frac{W_k^{(j-1)}(\vec{n}, [s_1, \cdots, s_p, r])}{Z_r + W_{pess}^{(j-1)}(\vec{n} - \sum_{g=1}^{p} \vec{e}_{s_g}, [r])} (n_r - \sum_{g=1}^{p} \delta_{rs_g}) \qquad (4\text{-}27)$$

where, for $1 \le l \le p$,

$$W_k^{(j)}(\vec{n}, [s_1, \cdots, s_p]) = (\prod_{g \ne l} L_{ks_g}) R_{ks_l}^{(j)}(\vec{n} - \sum_{g \ne l} \vec{e}_{s_g})$$

From the initial performance measure estimates of the previous section,

$$W_k^{(1)}(\vec{n}, [s_1, \cdots, s_p]) = \prod_{g=1}^{p} L_{ks_g}$$

The symbolic formulation of the multiple closed class PBH algorithm has both advantages and disadvantages in comparison to the non-symbolic formulation; many of these are qualitatively the same as with the single closed class PBH algorithm. In the next section, the computational cost of both formulations will be examined in detail.

4.3. Computational Cost

Table 4.3 summarizes results concerning the computational complexity of both the symbolic and the non-symbolic formulations of the PBH algorithm. The quantities given are the dominant terms in the space and time complexities of computing level i bounds for population level $\vec{N}$, assuming that $N_r \ge i$ for all $r \in C$. The cost of the linear programming solution for the $L_{sr\ LB}^h$ values has not been included here, as it is assumed to be dominated by the other computational requirements of the algorithm. (If this is not the case, $L_{sr\ LB}^h$ can be

simply chosen as the minimum, over the fixed-rate service centers visited by class r, of the L_{ks} values.) For MVA, the corresponding complexities are $4K|C|\prod_{r\in C}(N_r+1)$ for time, and $K|C|\prod_{r\neq max}(N_r+1)$ for space, where max denotes the class with the largest population.

	Non-symbolic	Symbolic
Time	$6K\|C\|\binom{\|C\|+i}{i}$	$2\|C\|\sum_{j=1}^{i}\binom{2\|C\|+i-j}{i-j+1}+2K\binom{\|C\|+i}{i}+2k\|C\|\binom{\|C\|+i}{i}$
Space	$K\|C\|\binom{\|C\|+i-1}{i}$	$\binom{2\|C\|+i-1}{i}+k\binom{\|C\|+i}{i}$

k = number of fixed-rate service centers for which secondary performance measures are required

The Computational Cost of the Multiple Closed Class PBH Algorithm

Table 4.3

The quantities given in table 4.3 for the non-symbolic formulation are derived directly from the space complexity ($K|C|$) and time complexity ($6K|C|$) of an application of the modified MVA equations. (This time complexity is $2K|C|$ higher than that for the unmodified equations due to the asymptotic bound computations.) Results for the symbolic formulation are based on a computational approach similar to that used for the symbolic formulation of the single closed class PBH algorithm. $W_{pess}^{(i)}(\vec{N},[s])$ values are calculated recursively, starting from $W_{pess}^{(1)}(\vec{n},[s_1,\cdots,s_p])$ values such that $\sum_{r\in C}(N_r-(n_r-\sum_{g=1}^{p}\delta_{rs_g}))=i$. At a typical step, all $W_{pess}^{(j)}(\vec{n},[s_1,\cdots,s_p])$ values (for some $j\leq i$) such that $\sum_{r\in C}(N_r-(n_r-\sum_{g=1}^{p}\delta_{rs_g}))=i-j+1$, are computed from all

$W_{pass}^{(j-1)}(\vec{m},[t_1,\cdots,t_q])$ values such that $\sum_{r\in C}(N_r-(m_r-\sum_{g=1}^{q}\delta_{rt_g}))=i-j+2$. There are $\binom{2|C|+i-j}{i-j+1}-\binom{|C|+i-j}{i-j+1}$ $W_{pass}^{(j)}(\vec{n},[s_1,\cdots,s_p])$ values to calculate for each value of j, which is of the order of $\binom{2|C|+i-j}{i-j+1}$. Each of these requires on the order of $2|C|$ operations, once the necessary $L^{[s_1,\cdots,s_p]}$ quantities have been computed (which require an additional order $2K\binom{|C|+i}{i}$ operations).

If bounds at multiple bound hierarchy levels are to be computed, or bounds for the same queueing network with different population levels, the symbolic formulation of the algorithm does not have to repeat the computation represented by the $2K\binom{|C|+i}{i}$ time complexity term, while the non-symbolic formulation must repeat everything. However, as the number of customer classes increases, so does the computational cost of the symbolic formulation relative to the non-symbolic formulation. The symbolic formulation is likely best applied to queueing networks with only a few classes and many service centers, when secondary performance measure bounds are not required.

4.4. Algorithm Accuracy

The accuracy of the multiple closed class PBH algorithm has been studied by applying the algorithm to 72 test queueing networks. Only closed queueing networks were chosen, since the accuracy of the algorithm when applied to networks with open classes depends on the accuracy of the algorithm when applied to the closed networks that must be analyzed to obtain the performance measures of the closed classes.

One third of the queueing networks had 2 classes, one third had 4 classes, and the remaining third had 6 classes. One half of the classes of each queueing network visited a delay service center with loading 20, and can be thought of as modelling interactive terminal user workloads. The remaining classes can be

thought of as modelling batch workloads.

One half of the queueing networks had 10 fixed-rate service centers, and the others had 20. The fixed-rate service center loadings of each class were chosen independently of the choices for each other class, by the same method that was used when constructing the test queueing networks for the single closed class PBH algorithm. Each loading was derived by sampling from a uniform distribution between 0 and 1 using a pseudo-random number generator, and then raising the result to a power p. Finally, the fixed-rate service center loadings of each class were normalized so that they summed to one. One third of the queueing networks were constructed using a p value of .5 for each class, one third using a p value of 1, and the remaining third using a p value of 2.

Each queueing network was analyzed with nine different population levels. The total number of customers in the classes modelling interactive workloads was chosen as either 20, 50 or 80, while the total number of customers in the batch workload classes was chosen as either 3, 6 or 9. Customers were divided evenly (except as constrained by a desire for integral population sizes) among the classes of each type.

Level 1, 3, 5 and 7 bounds were computed for each queueing network and each population level. These bounds were computed with each $L^h_{sr\ LB}$ value chosen as the minimum, over the fixed-rate service centers visited by class r, of the class s loadings. The major reason for not utilizing the $L^h_{sr\ LB}$ values that can be obtained by the linear programming method concerns the nature of the chosen test queueing networks. The linear programming solution for the $L^h_{sr\ LB}$ values is useful only when there is a correlation among the loadings of the various classes (for example, a common bottleneck exists). In the most extreme case, if the ordering of fixed-rate service centers from that with the highest to that with the lowest class s loading is the same for each class s, then the linear

programming solution for $L^h_{sr\ LB}$ is the average, rather than the minimum, fixed-rate service center loading of class s. Since the test queueing networks lack any such correlation, the $L^h_{sr\ LB}$ values obtained by the linear programming method would not be significantly better than the simple values chosen here. If the test queueing networks had such a correlation, our results would be significantly improved, partly due to the improved $L^h_{sr\ LB}$ values that would then be possible, and partly due to a diminished sensitivity to the value of p (see the discussion on this sensitivity that follows in this section).

Over each chosen grouping of tests, the maximum and the average maximal errors for system throughput and mean system residence time were computed for each bound level. Each average was computed by first averaging the errors over all of the classes of each queueing network, and then averaging over networks. Each maximum was computed by first finding the maximum error over all of the classes of each network, and then taking the maximum over networks. The outputs from analyzing several of the test queueing networks were examined in full. With each of these networks, the mean system residence time errors for the various classes differed by less than 3 percent. This implies that taking the maximum mean system residence time error over all of the classes of each network, and then averaging over networks, would yield results similar to the computed average errors. Similarly, averaging the mean system residence time errors over all of the classes of each network, and then taking the maximum over networks, would yield results similar to the computed maximum errors. The same statements cannot be made for the system throughput errors, since these errors are highly dependent on the delay service center loading of the class under consideration. (Note, however, that the system throughput error for a class is always bounded from above by the mean system residence time error for the class.)

Table 4.4 summarizes the results for the 648 combinations of queueing network and population level. Although the average maximal errors are acceptable, the maximum maximal errors are quite high. The size of this variation in algorithm accuracy can be accounted for by the results of table 4.9, which show the sensitivity of the multiple closed class PBH algorithm to the value of p that was used in generating the fixed-rate service center loadings. For p values of .5 or 1, quite acceptable accuracy is obtained. For a p value of 2, however, the algorithm accuracy is quite poor.

This can best be explained by considering the algorithm's symbolic formulation. The symbolic formulation uses the unmodified symbolic MVA equation (equation (4-25)) for the computation of the $R_s^{(i)}{}_{pess}(\vec{N})$ values, implying that the only error in these values (and therefore in the system throughput lower bounds and mean system residence time upper bounds) is due to errors in the initial estimates (equation (4-26)). High values of p, in conjunction with the lack of any correlation among the fixed-rate service center loadings of the various classes, tend to diminish $L^{[s_1,\ldots,s_p]}$ values and increase $\prod_{g=1}^{p} L_{s_g}^{h}{}_{UB}$ values, magnifying the effect of initial estimate errors. This sensitivity to p diminishes significantly when there are strong correlations among the loadings.

Tables 4.5 to 4.8, like table 4.9, consider only the level 5 bounds. Table 4.5 shows that the average and maximum of the maximal errors for both system throughput and mean system residence time increase slightly as the number of customer classes increases. Tables 4.6 to 4.8 show that the average and maximum maximal errors increase as the queueing network becomes more congested. This last effect is quite significant, and can again be explained by considering the algorithm's symbolic formulation. As the queueing network population increases, the summation in the second term in equation (4-26) increases. This magnifies the effect of the errors induced by the pessimistic heap

Bound Level	Maximal Errors (%)			
	$R_c(\vec{N})$ Aver.	$R_c(\vec{N})$ Max.	$X_c(\vec{N})$ Aver.	$X_c(\vec{N})$ Max.
1	36.7	88.4	19.2	85.4
3	17.4	81.3	9.2	81.3
5	10.7	79.8	5.7	79.8
7	7.6	79.1	4.1	79.1

Maximal Errors for a Range of Bound Levels

Table 4.4

Number of Classes	Maximal Errors (Level 5 Bounds) (%)			
	$R_c(\vec{N})$ Aver.	$R_c(\vec{N})$ Max.	$X_c(\vec{N})$ Aver.	$X_c(\vec{N})$ Max.
2	8.4	62.0	4.4	62.0
4	11.6	69.8	6.3	69.8
6	12.1	79.8	6.5	79.8

Maximal Errors for Various Numbers of Closed Classes

Table 4.5

Number of Fixed-Rate Service Centers	Maximal Errors (Level 5 Bounds) (%)			
	$R_c(\vec{N})$ Aver.	$R_c(\vec{N})$ Max.	$X_c(\vec{N})$ Aver.	$X_c(\vec{N})$ Max.
10	16.9	79.8	9.2	79.8
20	4.4	46.9	2.2	35.2

Maximal Errors for two Different Queueing Network Sizes

Table 4.6

Interactive Workload Intensity	Maximal Errors (Level 5 Bounds) (%)			
	$R_s(\vec{N})$ Aver.	$R_s(\vec{N})$ Max.	$X_s(\vec{N})$ Aver.	$X_s(\vec{N})$ Max.
20	3.6	42.2	1.6	32.4
50	10.0	63.6	5.1	61.7
80	18.4	79.8	10.4	79.8

Maximal Errors for a Range of Total Interactive Workload Intensities

Table 4.7

Batch Workload Intensity	Maximal Errors (Level 5 Bounds) (%)			
	$R_s(\vec{N})$ Aver.	$R_s(\vec{N})$ Max.	$X_s(\vec{N})$ Aver.	$X_s(\vec{N})$ Max.
3	6.3	74.1	3.4	74.1
6	10.7	77.6	5.7	77.6
9	15.1	79.8	8.0	79.8

Maximal Errors for a Range of Total Batch Workload Intensities

Table 4.8

Loading Skewness (value of p)	Maximal Errors (Level 5 Bounds) (%)			
	$R_s(\vec{N})$ Aver.	$R_s(\vec{N})$ Max.	$X_s(\vec{N})$ Aver.	$X_s(\vec{N})$ Max.
0.5	3.0	22.9	1.6	19.9
1.0	8.4	48.5	4.3	46.5
2.0	20.6	79.8	11.2	79.8

Maximal Errors for a Range of p Values

Table 4.9

allocation, which is reflected in this equation by the factor of $\prod_{g=1}^{p} L_{s_g}^{h}$ $_{UB}$.

In summary, the PBH algorithm was of acceptable accuracy for most of the test queueing networks. However, for some of the networks the accuracy was quite poor. The following queueing network characteristics were found to be important in determining the algorithm accuracy: (1) the degree of correlation among the loadings of the various classes (the more dissimilar the classes are, the larger the errors); (2) the skewness of the loadings (the greater the skewness of the service center loadings of a class, the larger the errors); and (3) the degree of congestion in the queueing network (the greater the congestion, the larger the errors).

4.5. An Error-Bounded Approximation Algorithm

4.5.1. Motivation

In an error-bounded approximation algorithm, a major portion of the computational effort is expended on point estimates. Bounds on the point estimate errors, if desired, can be provided after the point estimates have been obtained. The major advantages of such an algorithm are, first of all, the greater point estimate accuracy it typically attains (with respect to a bounding algorithm), and, secondly, the fact that error bounds are optional. If a number of similar queueing networks are being analyzed, the assumption can be made that the point estimate error bounds for each network are similar, allowing the computational expense to be reduced by only computing error bounds for the networks of most interest.

In this section, an error-bounded approximation algorithm based on MVA is presented. The point estimates produced (and the exact solution) can be bounded by the multiple closed class PBH algorithm, as modified by removing from consideration the asymptotic bounds $\alpha_s^{(i)}(\vec{N})$. Given a point estimate A, a

lower bound L, and an upper bound U ($L \leq A \leq U$), the magnitude of the percent relative error in A can be bounded by $max(\frac{A-L}{L} * 100\%, \frac{U-A}{U} * 100\%)$.

4.5.2. Algorithm Structure

For the multiple closed class PBH algorithm, it was necessary to modify the MVA equations. If these equations are not modified, the performance measure estimates produced are more accurate, although their relationship to the exact performance measures is unknown. The error-bounded approximation algorithm that is suggested here is based on the use of the unmodified MVA equations. As usual, only closed queueing networks need be explicitly considered when describing the algorithm structure.

Subscripting approximate values by *appr*, the level i approximations ($i \geq 1$) are defined by, for each $s \in C$,

$$R_{ks}^{(i)}{}_{appr}(\vec{N}) = L_{ks}(1 + \sum_{r \in C} n_{ks}^{(i-1)}{}_{appr}(\vec{N} - \vec{e}_s)) \qquad 1 \leq k \leq K$$

$$n_{ks}^{(i)}{}_{appr}(\vec{N}) = \frac{R_{ks}^{(i)}{}_{appr}(\vec{N})}{Z_s + R_s^{(i)}{}_{appr}(\vec{N})} N_s \qquad 1 \leq k \leq K$$

and

$$R_s^{(i)}{}_{appr}(\vec{N}) = \sum_{k=1}^{K} R_{ks}^{(i)}{}_{appr}(\vec{N})$$

The initial performance measure estimates (level 0 approximations) are chosen so that they are bounded by the initial estimates of the multiple closed class PBH algorithm, and so that $n_{ks}^{(0)}{}_{appr}(\vec{N}) \leq n_{js}^{(0)}{}_{appr}(\vec{N})$ if $L_{kt} \leq L_{jt}$ for all $t \in C$ ($1 \leq j \leq K, 1 \leq k \leq K$). One suitable choice of initial estimates sets $R_s^{(0)}{}_{appr}(\vec{N})$ to the harmonic mean of $R_s^{(0)}{}_{pess}(\vec{N})$ and $R_s^{(0)}{}_{opt}(\vec{N})$ (which were chosen in subsection 4.1.2 as $b_s(\vec{N})$ and $a_s^{(0)}(\vec{N})$, respectively), for all $s \in C$. The mean system residence time estimate for each class s is divided among the fixed-rate service centers in direct proportion to the class s loadings. Little's equation is then used to compute the initial mean queue lengths.

When the asymptotic bounds are not utilized in the multiple closed class PBH algorithm, setting the initial heaps to zero results in a computation identical to that when using the unmodified MVA equations (as can be verified from equations (4-1) to (4-6)). For the purposes of showing that the level i approximations produced by the error-bounded approximation algorithm are bounded by the level i bounds yielded by the PBH algorithm, the error-bounded approximation algorithm can be thought of as using the PBH algorithm equations (with the $a_s^{(i)}(\vec{N})$ bounds removed from consideration), with initial heap estimates of zero. The claim then follows from the choice of initial estimates in the error-bounded approximation algorithm and the PBH algorithm, lemmas 4.1 to 4.3 (where lemma 4.3 is modified to reflect the removal of the $a_s^{(i)}(\vec{N})$ bounds, and the primed quantities are taken as those of the error-bounded approximation algorithm), and theorem 4.2 (where the exact performance measures are replaced by those of the error-bounded approximation algorithm).

4.5.3. An Example

Table 4.10 gives level 2 and level 5 approximations for the example queueing network of table 4.1, for a class T arrival rate of .1, a population of 3 class B customers, and a population of 35 class I customers. As in table 4.2, the performance measures considered are the mean system residence time of each class and the system throughputs of classes B and I (with time unit choice such that the sum of the fixed-rate service center loadings of each class equals one), the utilizations by classes B and I and the total utilization of each fixed-rate service center, and the mean queue length of each class at each fixed-rate service center. The initial estimates used were those suggested in the previous subsection.

System Throughput Approximations					
Class *B*			Class *I*		
Level 2	Level 5	Exact	Level 2	Level 5	Exact
1.46	1.44	1.44	1.07	1.06	1.06

Mean System Residence Time Approximations								
Class *T*			Class *B*			Class *I*		
Level 2	Level 5	Exact	Level 2	Level 5	Exact	Level 2	Level 5	Exact
3.00	3.08	3.09	2.05	2.08	2.08	2.75	2.88	2.90

Fixed-Rate Service Center Utilization Approximations									
Service Center	Class *B*			Class *I*			Total over all classes		
	Level 2	Level 5	Exact	Level 2	Level 5	Exact	Level 2	Level 5	Exact
1	.14	.13	.13	.21	.21	.21	.85	.85	.85
2	.07	.07	.07	.11	.11	.11	.57	.57	.57
3	.18	.17	.17	.05	.05	.05	.57	.57	.57
4	.04	.04	.04	.27	.27	.27	.63	.63	.63
5	.23	.23	.23	.06	.06	.06	.45	.45	.45
6	.26	.25	.25	.18	.18	.18	.56	.55	.55
7	.34	.33	.33	.15	.15	.15	.59	.58	.58
8	.22	.21	.21	.03	.03	.03	.31	.31	.31

Fixed-Rate Service Center Mean Queue Length Approximations									
Service Center	Class *T*			Class *B*			Class *I*		
	Level 2	Level 5	Exact	Level 2	Level 5	Exact	Level 2	Level 5	Exact
1	2.61	2.80	2.82	.59	.64	.65	1.02	1.15	1.17
2	.94	.93	.93	.15	.15	.15	.25	.25	.25
3	.77	.76	.76	.37	.36	.35	.12	.12	.12
4	.84	.84	.84	.10	.10	.10	.68	.69	.69
5	.28	.28	.28	.38	.37	.36	.12	.11	.11
6	.25	.25	.25	.49	.48	.48	.38	.38	.38
7	.22	.22	.22	.64	.62	.62	.32	.32	.32
8	.09	.08	.08	.29	.28	.28	.05	.05	.05

Level 2 and Level 5 Approximations for an Example Queueing Network

Table 4.10

4.6. Modifications to Allow Load Dependence

The MVA algorithm, and therefore an appropriate modification of the PBH algorithm, can also be applied to queueing networks containing load-dependent service centers (at which both the service rate received by each customer and the total service rate vary with the queue composition). This section treats a common type of load-dependent service center that is characterized by a total service rate that depends only on the total number of customers (regardless of class) that are present. Only closed queueing networks will be considered; for queueing networks with open classes, the closed class performance measures can be obtained by solving a closed network in which the open classes have been removed (in a similar fashion as for queueing networks with only fixed-rate and delay service centers), although it is somewhat more complicated to then obtain the open class performance measures from this solution [Sauer 83].

The only modification to the MVA equations that is necessitated by the presence of load-dependent service centers is the addition of special equations to compute the mean residence times at these service centers. The mean residence time equation for a load-dependent service center k and a class s is

$$R_{ks}(\vec{N}) = \frac{L_{ks}(c_k)}{c_k}(1+\sum_{r\in C} n_{kr}(\vec{N}-\vec{e}_s)) + \sum_{m=1}^{c_k-1}(L_{ks}(m)-L_{ks}(c_k)\frac{m}{c_k})p_k(m-1\,|\,\vec{N}-\vec{e}_s)$$

where

$$p_k(m\,|\,\vec{N}) = \sum_{r\in C} L_{kr}(m)X_r(\vec{N})p_k(m-1\,|\,\vec{N}-\vec{e}_r) \qquad 1\le m\le c_k-2$$

and, for any $r\in C$ such that $N_r\ge 1$,

$$p_k(0\,|\,\vec{N}) = p_k(0\,|\,\vec{N}-\vec{e}_r)\frac{X_r(\vec{N})}{X_r^{-k}(\vec{N})}$$

In these equations, c_k denotes the *capacity* of service center k (the total service rate is constant when c_k or more customers are present), $p_k(m\,|\,\vec{N})$ denotes the probability that m customers are present at service center k, and

the superscript $-k$ denotes performance measures for the queueing network with service center k removed.

Due to the expression used for $p_k(0|\vec{N})$, the MVA algorithm must also compute the performance measures for each of the queueing networks resulting from the removal of a load-dependent service center. Each of these queueing networks, if it contains load-dependent service centers, similarly requires the solution of additional queueing networks. An alternative approach is to calculate $p_k(0|\vec{N})$ from the normalizing condition $\sum_{m=0}^{N} p_k(m|\vec{N})=1$; however, this method has been found to be numerically unstable [Chandy & Sauer 80, Reiser 81].

For clarity, when describing the modifications to the multiple closed class PBH algorithm necessitated by load-dependent service centers, the assumption will be made that $\frac{L_{ks}(m)}{m} \geq \frac{L_{ks}(m+1)}{m+1}$ for all $m \geq 1$. (This relation corresponds to a non-decreasing total service rate, and is usually satisfied in the computer system modelling context.) The PBH equations used to derive mean residence time values for a load-dependent service center k are of the same form as those used by MVA. For $i \geq 1$, and each class $s \in C$,

$$R_{ks}^{(i)}(\vec{N}) = \frac{L_{ks}(c_k)}{c_k}\left(1+\sum_{r\in C} n_{kr}^{(i-1)}(\vec{N}-\vec{e}_s)\right) + \sum_{m=1}^{c_k-1}\left(L_{ks}(m)-L_{ks}(c_k)\frac{m}{c_k}\right)p_k^{(i-1)}(m-1|\vec{N}-\vec{e}_s) \qquad (4\text{-}28)$$

where

$$p_k^{(i)}(m|\vec{N}) = \sum_{r\in C} L_{kr}(m) X_{r\,pess}^{(i)}(\vec{N}) p_k^{(i-1)}(m-1|\vec{N}-\vec{e}_r) \qquad 1\leq m\leq c_k-2 \qquad (4\text{-}29)$$

and, for any $r \in C$ such that $N_r \geq 1$,

$$p_k^{(i)}(0|\vec{N}) = p_k^{(i-1)}(0|\vec{N}-\vec{e}_r)\frac{X_{r\,pess}^{(i)}(\vec{N})}{X_{r\,opt}^{-k\,(i)}(\vec{N})} \qquad (4\text{-}30)$$

In this last equation, class r is chosen to be a class with a large delay service

center loading (if such a class exists), since the system throughput of such a class, when calculated using Little's equation, is quite insensitive to errors in the mean system residence time of the class. Suitable initial performance measure estimates corresponding to equations (4-28) to (4-30) are

$$R_{ks}^{(0)}(\vec{N}) = n_{ks}^{(0)}(\vec{N}) = 0$$
$$p_k^{(0)}(m \mid \vec{N}) = 0 \qquad 1 \le m \le c_k - 2$$

and

$$p_k^{(0)}(0 \mid \vec{N}) = max\left[0\ ,\ 1 - \sum_{r \in C} X_{opt}^{(0)}(\vec{N}) L_{kr}(1)\right]$$

Note that $X_r(\vec{N}) L_{kr}(1) \ge U_{kr}(\vec{N})$ (as follows from the expression for $U_{kr}(\vec{N})$ that is used in the Convolution algorithm [Zahorjan 80]), and that therefore $p_k^{(0)}(0 \mid \vec{N}) \le p_k(0 \mid \vec{N})$.

The multiple closed class PBH algorithm must obtain performance measures for each of the queueing networks resulting from the removal of a load-dependent service center, as in MVA. (Each of these queueing networks, if it contains load-dependent service centers, similarly requires performance measures for additional queueing networks.) In particular, the PBH algorithm requires level i optimistic class r system throughputs, for each of the queueing networks with a load-dependent service center removed, when calculating level i performance measures for the original queueing network. This may be computationally infeasible if there are many load-dependent service centers; in these cases, rather than using level i optimistic system throughputs, level 0 throughputs can be utilized (which do not require the analysis of additional queueing networks). Although the resulting algorithm loses some accuracy, as well as the property that the level i performance measures are exact for $i=N$, the computational savings can be large.

The remaining modifications necessitated by the presence of load-dependent service centers are to the computation of $R_s^{(i)}{}_{pess}(\vec{N})$, $\alpha_s^{(i)}(\vec{N})$, and

$b_s(\vec{N})$, and to the choices of $L^h_{sr\ UB}$ and $L^h_{sr\ LB}$. For each load-dependent service center k, the additional term

$$(1-\sum_{m=1}^{c_k-1} p_k^{(i-1)}(m-1\,|\,\vec{N}-\vec{e}_s))(L_{ks}(1)-\frac{L_{ks}(c_k)}{c_k})$$

must be added to $R^{(i)}_{s\,pess}(\vec{N})$. This compensates for any possible underestimate of $p_k(0\,|\,\vec{N}-\vec{e}_s)$ (as can be seen from equation (4-28)). In the defining equation for $a_s^{(i)}(\vec{N})$ (equation (4-7)), for each load-dependent service center k $\min_m[L_{ks}(m)]$ is added to the bound of 1 (since $R_{ks}(\vec{N}) \geq \min_m[L_{ks}(m_i)]$). In the other bound in this equation, each load-dependent service center is treated exactly as is each fixed-rate service center, except that $\frac{L_{kr}(c_k)}{c_k}$ is used for a load-dependent service center where L_{kr} is used for a fixed-rate service center. This modification is correct, since, for a load-dependent service center k, $X_r(\vec{N})\frac{L_{kr}(c_k)}{c_k} \leq U_{kr}(\vec{N})$ (as follows directly from the expression for $U_{kr}(\vec{N})$ used in the Convolution algorithm). For each load-dependent service center k, $L_{ks}(1)$ must be added to the expression giving $b_s(\vec{N})$ in terms of $b_r(\vec{N})$ for all $r \in C$. This modification follows from the fact that the mean residence time at a load-dependent service center k is maximized for $p_k(0\,|\,\vec{N}-\vec{e}_s)$ maximized (when this probability is viewed as independent of all other probabilities), and from the construction of $L^h_{sr\ UB}$.

$L^h_{sr\ UB}$ is chosen as the maximum of the value obtained when not considering the load-dependent service centers, and the maximum, over the load-dependent service centers k visited by class r, of $\max_{m\geq 2}\left[\frac{L_{ks}(m)-\frac{L_{ks}(c_k)}{c_k}}{m-1}\right]$. This latter expression is derived from equation (4-28) by supposing that the class r heap is due solely to an underestimate of $p_k(m-1\,|\,\vec{N}-\vec{e}_s)$, and then solving for the effect of this underestimate on $R^{(i)}_{ks}(\vec{N})$. $L^h_{sr\ LB}$ is chosen as the minimum of

the value obtained when not considering the load-dependent service centers, and the minimum, over the load-dependent service centers k visited by class r, of $\frac{L_{ks}(c_k)}{c_k}$. $\frac{L_{ks}(c_k)}{c_k}$ is used since the most optimistic possible allocation of a heap to service center k inflates the probabilities $p_k(m-1|\vec{N}-\vec{e}_s)$ for $m \geq c_k$ (as can be verified from equation (4-28)). Improvements to the $L^h_{sr\ UB}$ and $L^h_{sr\ LB}$ choices are possible; however, they create a dependence between the values of $L^h_{sr\ UB}$ and $L^h_{sr\ LB}$ and the heap size. The resulting increase in algorithm complexity is likely unjustified.

In lemma 4.5, it is proved that the character of initial estimates is preserved by the multiple closed class PBH equations, as modified to allow for load-dependent service centers. In addition to preserving the relations that were preserved previously, $p_k^{(i)}(m|\vec{N}) \leq p_k(m|\vec{N})$ for $0 \leq m \leq c_k - 2$, $i \geq 0$, is also preserved for each load-dependent service center k. Lemma 4.5, in conjunction with the appropriate character of the chosen initial estimates, implies that theorem 4.2 holds, with a few modifications, when load-dependent service centers are allowed. In the mean queue length and mean residence time bounds for each load-dependent service center k, A_k should be replaced by 1. (For fixed-rate service centers, A_k is computed as if the load-dependent service centers were not present.) $\frac{L_{ks}(c_k)}{c_k}$ should be used in the lower bound on the class s utilization of each load-dependent service center k, while $L_{ks}(1)$ should be used in the upper bound. The upper bound should be used in conjunction with the bound given by $1-p_k^{(i)}(0|\vec{N})$. Finally, note that tertiary performance measure bounds for each load-dependent service center are also available, as the algorithm yields lower bounds on the $p_k(m|\vec{N})$ quantities for $0 \leq m \leq c_k - 2$.

APPENDIX 4.A

This appendix states and proves lemmas of relevance to chapter four.

Lemma 4.1:

For all populations $\vec{n}$ such that $|\vec{n}|=n$, for some n, suppose that two mean queue length allocations with associated pessimistic heaps, $[n_{ks}^{v(0)}(\vec{n})]_{1\le k\le K, s\in C}$ $(0\le n_{ks}^{v(0)}(\vec{n})\le n_{ks}(\vec{n}))$ with $n_{s\ pess}^{hv(0)}(\vec{n})$ (for all $s\in C$), and $[n_{ks}^{v'(0)}(\vec{n})]_{1\le k\le K, s\in C}$ $(0\le n_{ks}^{v'(0)}(\vec{n})\le n_{ks}(\vec{n}))$ with $n_{s\ pess}^{hv'(0)}(\vec{n})$ (for all $s\in C$), are such that, for each $s\in C$,

$$n_{ks}^{v(0)}(\vec{n}) \le n_{ks}^{v'(0)}(\vec{n}) \qquad 1\le k\le K$$

and

$$n_{s\ pess}^{hv(0)}(\vec{n})+\sum_{k=1}^{K} n_{ks}^{v(0)}(\vec{n}) \ge n_{s\ pess}^{hv'(0)}(\vec{n})+\sum_{k=1}^{K} n_{ks}^{v'(0)}(\vec{n})$$

Then for $s\in C$, $i\ge 0$, and population $\vec{N}$ such that $|\vec{N}| = n+i$,

$$n_{ks}^{v(i)}(\vec{N}) \le n_{ks}^{v'(i)}(\vec{N}) \qquad 1\le k\le K \tag{4.1-1}$$

and

$$n_{s\ pess}^{hv(i)}(\vec{N})+\sum_{k=1}^{K} n_{ks}^{v(i)}(\vec{N}) \ge n_{s\ pess}^{hv'(i)}(\vec{N})+\sum_{k=1}^{K} n_{ks}^{v'(i)}(\vec{N}) \tag{4.1-2}$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by applying the equations of the multiple closed class PBH algorithm starting from values superscripted by $v(0)$ and $v'(0)$, respectively. (Optimistic heaps are not needed to establish this claim and have therefore been disregarded.)

Proof:

The proof is by induction on i.

By assumption, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ $(i\ge 0)$. From relation (4-6), relation (4.1-2) will be established if it can be shown that, for each $s\in C$,

$$R_{s\ pess}^{v(i+1)}(\vec{N}) \ge R_{s\ pess}^{v'(i+1)}(\vec{N}) \tag{4.1-3}$$

which reduces to, using equation (4-3),

$$\sum_{k=1}^{K} R_{ks}^{v(i+1)}(\vec{N}) + \sum_{r\in C} L_{sr}^{h}\ {}_{VB}n_{r\ pess}^{hv(i)}(\vec{N}-\vec{e}_s) \ge \sum_{k=1}^{K} R_{ks}^{v'(i+1)}(\vec{N}) + \sum_{r\in C} L_{sr}^{h}\ {}_{VB}n_{r\ pess}^{hv'(i)}(\vec{N}-\vec{e}_s)$$

From equation (4-1), this relation can be expanded to yield, after regrouping terms and canceling $\sum_{k=1}^{K} L_{ks}$ from both sides,

$$\sum_{r\in C} L_{sr}^{h}\ {}_{VB}(n_{r\ pess}^{hv(i)}(\vec{N}-\vec{e}_s)-n_{r\ pess}^{hv'(i)}(\vec{N}-\vec{e}_s)) \ge \sum_{r\in C}\sum_{k=1}^{K} L_{ks}(n_{kr}^{v'(i)}(\vec{N}-\vec{e}_s)-n_{kr}^{v(i)}(\vec{N}-\vec{e}_s))$$

From the inductive hypothesis on relation (4.1-1), each term in the double summation on the right-hand side of this last relation is non-negative. These terms represent an allocation of a heap for each class r, given by $\sum_{k=1}^{K} n_{kr}^{v'(i)}(\vec{N}-\vec{e}_s) - \sum_{k=1}^{K} n_{kr}^{v(i)}(\vec{N}-\vec{e}_s)$, to the fixed-rate service centers. Since, by

construction, $L^h_{sr\ UB}$ is an upper bound on the mean class s loading when encountering customers from the class r heap, the relation holds if it holds with L_{ks} replaced by $L^h_{sr\ UB}$. Making this substitution, separating the terms corresponding to each class, and canceling $L^h_{sr\ UB}$, results in relation (4.1-2) for i, which holds from the inductive hypothesis. This establishes relation (4.1-2) for $i+1$.

From relations (4-4) and (4-1), and the inductive hypothesis on relation (4.1-1), relation (4.1-3) also establishes relation (4.1-1) for $i+1$. This establishes the lemma for $i+1$, and, by induction, for all $i \geq 0$. ∎

Lemma 4.2:

For all populations $\vec{n}$ such that $|\vec{n}|=n$, for some n, suppose that two mean queue length allocations with associated pessimistic heaps, $[n^{v(0)}_{ks}(\vec{n})]_{1\leq k\leq K, s\in C}$ $(0\leq n^{v(0)}_{ks}(\vec{n})\leq n_{ks}(\vec{n}))$ with $n^{hv(0)}_{s\ pess}(\vec{n})$ (for all $s\in C$), and $[n^{v'(0)}_{ks}(\vec{n})]_{1\leq k\leq K, s\in C}$ $(0\leq n^{v'(0)}_{ks}(\vec{n})\leq n_{ks}(\vec{n}))$ with $n^{hv'(0)}_{s\ pess}(\vec{n})$ (for all $s\in C$), are such that, for each $s\in C$,

$$n^{v(0)}_{ks}(\vec{n}) \leq n^{v'(0)}_{ks}(\vec{n}) \qquad 1\leq k\leq K$$

$$n^{hv(0)}_{s\ pess}(\vec{n}) + \sum_{k=1}^{K} n^{v(0)}_{ks}(\vec{n}) \geq n^{hv'(0)}_{s\ pess}(\vec{n}) + \sum_{k=1}^{K} n^{v'(0)}_{ks}(\vec{n})$$

and, for fixed-rate service centers k and j such that $L_{kt}\leq L_{jt}$ for all classes t,

$$n^{v'(0)}_{ks}(\vec{n}) - n^{v(0)}_{ks}(\vec{n}) \leq n^{v'(0)}_{js}(\vec{n}) - n^{v(0)}_{js}(\vec{n})$$

Then for $s\in C$, $i\geq 0$, and population $\vec{N}$ such that $|\vec{N}| = n+i$,

$$n^{v'(i)}_{ks}(\vec{N}) - n^{v(i)}_{ks}(\vec{N}) \leq n^{v'(i)}_{js}(\vec{N}) - n^{v(i)}_{js}(\vec{N}) \qquad (4.2\text{-}1)$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by applying the equations of the multiple closed class PBH algorithm starting from values superscripted by $v(0)$ and $v'(0)$, respectively. (Optimistic heaps are not needed to establish this claim and have therefore been disregarded.)

Proof:

The proof is by induction on i.

By assumption, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ $(i\geq 0)$. Regrouping terms in relation (4.2-1) and utilizing equation (4-4) yields, for each $s\in C$,

$$\frac{N_s}{Z_s + R^{v(i+1)}_{s\ pess}(\vec{N})}\left(R^{v(i+1)}_{js}(\vec{N}) - R^{v(i+1)}_{ks}(\vec{N})\right) \leq \frac{N_s}{Z_s + R^{v'(i+1)}_{s\ pess}(\vec{N})}\left(R^{v'(i+1)}_{js}(\vec{N}) - R^{v'(i+1)}_{ks}(\vec{N})\right)$$

From relation (4.1-3) of the previous lemma, this last relation holds if it holds with $R^{v(i+1)}_{s\ pess}(\vec{N})$ replaced by $R^{v'(i+1)}_{s\ pess}(\vec{N})$. Making this substitution, simplifying, and utilizing relation (4-1) results in

$$L_{js}\left(1+\sum_{r\in C} n^{v(i)}_{jr}(\vec{N}-\vec{e}_s)\right) - L_{ks}\left(1+\sum_{r\in C} n^{v(i)}_{kr}(\vec{N}-\vec{e}_s)\right) \leq$$

$$L_{js}\left(1+\sum_{r\in C} n^{v'(i)}_{jr}(\vec{N}-\vec{e}_s)\right) - L_{ks}\left(1+\sum_{r\in C} n^{v'(i)}_{kr}(\vec{N}-\vec{e}_s)\right)$$

Since $L_{js}\geq L_{ks}$ by assumption, and $n^{v'(i)}_{kr}(\vec{N}-\vec{e}_s)\geq n^{v(i)}_{kr}(\vec{N}-\vec{e}_s)$ for all $r\in C$ from relation (4.1-1) of the previous lemma, this last relation holds if it holds with L_{ks} replaced by L_{js}. Making this substitution, simplifying, and separating the

terms corresponding to each class r yields

$$n_{kr}^{v'(i)}(\vec{N}-\vec{e}_s)-n_{kr}^{v(i)}(\vec{N}-\vec{e}_s) \le n_{jr}^{v'(i)}(\vec{N}-\vec{e}_s)-n_{jr}^{v(i)}(\vec{N}-\vec{e}_s)$$

which holds from the inductive hypothesis. This establishes the lemma for $i+1$, and, by induction, for all $i \ge 0$. ∎

Lemma 4.3:

For all populations $\vec{n}$ such that $|\vec{n}|=n$, for some n, suppose that two mean queue length allocations with associated pessimistic and optimistic heaps, $[n_{ks}^{v(0)}(\vec{n})]_{1\le k\le K, s\in C}$ $(0\le n_{ks}^{v(0)}(\vec{n})\le n_{ks}(\vec{n}))$ with $n_{s\ pess}^{hv(0)}(\vec{n})$, $n_{s\ opt}^{hv(0)}(\vec{n})$ (for all $s\in C$), and $[n_{ks}^{v'(0)}(\vec{n})]_{1\le k\le K, s\in C}$ $(0\le n_{ks}^{v'(0)}(\vec{n})\le n_{ks}(\vec{n}))$ with $n_{s\ pess}^{hv'(0)}(\vec{n})$, $n_{s\ opt}^{hv'(0)}(\vec{n})$ (for all $s\in C$), are such that, for each $s\in C$,

$$n_{ks}^{v(0)}(\vec{n}) \le n_{ks}^{v'(0)}(\vec{n}) \qquad 1\le k\le K$$

$$n_{s\ pess}^{hv(0)}(\vec{n})+\sum_{k=1}^{K} n_{ks}^{v(0)}(\vec{n}) \ge n_{s\ pess}^{hv'(0)}(\vec{n})+\sum_{k=1}^{K} n_{ks}^{v'(0)}(\vec{n})$$

and

$$n_{s\ opt}^{hv(0)}(\vec{n})+\sum_{k=1}^{K} n_{ks}^{v(0)}(\vec{n}) \le n_{s\ opt}^{hv'(0)}(\vec{n})+\sum_{k=1}^{K} n_{ks}^{v'(0)}(\vec{n})$$

Then for $s\in C$, $i\ge 0$, and population $\vec{N}$ such that $|\vec{N}| = n+i$,

$$n_{s\ opt}^{hv(i)}(\vec{N})+\sum_{k=1}^{K} n_{ks}^{v(i)}(\vec{N}) \le n_{s\ opt}^{hv'(i)}(\vec{N})+\sum_{k=1}^{K} n_{ks}^{v'(i)}(\vec{N}) \qquad (4.3\text{-}1)$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by applying the equations of the multiple closed class PBH algorithm starting from values superscripted by $v(0)$ and $v'(0)$, respectively.

Proof:

The proof is by induction on i.

By assumption, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ $(i\ge 0)$. From relation (4-5), relation (4.3-1) will be established if it can be shown that, for each $s\in C$,

$$R_{s\ opt}^{v(i+1)}(\vec{N}) \le R_{s\ opt}^{v'(i+1)}(\vec{N}) \qquad (4.3\text{-}2)$$

Since lemma 4.1 implies that $a_s^{v(i+1)}(\vec{N}) \le a_s^{v'(i+1)}(\vec{N})$, from equation (4-2) relation (4.3-2) will be shown if it can be shown that

$$\sum_{k=1}^{K} R_{ks}^{v(i+1)}(\vec{N}) + \sum_{r\in C} L_{sr}^{h}\ {}_{LB} n_{r\ opt}^{hv(i)}(\vec{N}-\vec{e}_s) \le \sum_{k=1}^{K} R_{ks}^{v'(i+1)}(\vec{N}) + \sum_{r\in C} L_{sr}^{h}\ {}_{LB} n_{r\ opt}^{hv'(i)}(\vec{N}-\vec{e}_s)$$

From equation (4-1), this relation can be expanded to yield, after regrouping terms and canceling $\sum_{k=1}^{K} L_{ks}$ from both sides,

$$\sum_{r\in C} L_{sr}^{h}\ {}_{LB}(n_{r\ opt}^{hv(i)}(\vec{N}-\vec{e}_s)-n_{r\ opt}^{hv'(i)}(\vec{N}-\vec{e}_s)) \le \sum_{r\in C}\sum_{k=1}^{K} L_{ks}(n_{kr}^{v'(i)}(\vec{N}-\vec{e}_s)-n_{kr}^{v(i)}(\vec{N}-\vec{e}_s))$$

From lemma 4.1, each term in the double summation on the right-hand side of this last relation is non-negative. These terms represent an allocation of a heap for each class r, given by $\sum_{k=1}^{K} n_{kr}^{v'(i)}(\vec{N}-\vec{e}_s) - \sum_{k=1}^{K} n_{kr}^{v(i)}(\vec{N}-\vec{e}_s)$, to the fixed-

rate service centers. Since, by construction, $L^h_{sr\ LB}$ is a lower bound on the mean class s loading when encountering customers from the class r heap, the relation holds if it holds with L_{ks} replaced by $L^h_{sr\ LB}$ Making this substitution, separating the terms corresponding to each class, and canceling $L^h_{sr\ LB}$, results in relation (4.3-1) for i, which holds from the inductive hypothesis. This establishes the lemma for $i+1$, and, by induction, for all $i \geq 0$. ∎

Lemma 4.4:

For all $i \geq 0$, $s \in C$ and $\vec{N}$ such that $N_s \geq 1$,

$$b_s^{(i)}(\vec{N}) \geq R_s(\vec{N}) \tag{4.4-1}$$

where, for $i \geq 1$,

$$b_s^{(i)}(\vec{N}) = 1 + \sum_{r \in C} \frac{b_r^{(i-1)}(\vec{N})}{Z_r + b_r^{(i-1)}(\vec{N})} (N_r - \delta_{rs}) L^h_{sr\ UB}$$

and

$$b_s^{(0)}(\vec{N}) = N$$

Proof:

The lemma will be shown if it can be shown that

$$b_s^{(i)}(\vec{N}) \geq R_s(\vec{n}) \tag{4.4-2}$$

for all $\vec{n} \leq \vec{N}$. The proof is by induction on i.

Relation (4.4-2) trivially holds for $i=0$. Assume that the relation holds for $i \geq 0$, and consider the relation for $i+1$. From Little's equation, and a regrouping of terms, the MVA mean residence time equation yields

$$R_s(\vec{n}) = 1 + \sum_{r \in C} \sum_{k=1}^{K} L_{ks} \frac{R_{kr}(\vec{n} - \vec{e}_s)}{Z_r + R_r(\vec{n} - \vec{e}_s)} (n_r - \delta_{rs})$$

From the previous equation and the definition of $b_s^{(i+1)}(\vec{N})$, relation (4.4-2) will be shown for $i+1$ if it can be shown that, for each class r,

$$\frac{b_r^{(i)}(\vec{N})}{Z_r + b_r^{(i)}(\vec{N})} L^h_{sr\ UB} \geq \sum_{k=1}^{K} L_{ks} \frac{R_{kr}(\vec{n} - \vec{e}_s)}{Z_r + R_r(\vec{n} - \vec{e}_s)}$$

This last relation holds from the inductive hypothesis and the construction of $L^h_{sr\ UB}$. By induction, this establishes relation (4.4-2), and therefore the lemma, for all $i \geq 0$. ∎

Lemma 4.5:

For all populations $\vec{n}$ such that $|\vec{n}| = n$, for some n, and each queueing network that must be analyzed by the multiple closed class PBH algorithm as extended to allow for load dependence, suppose that two choices of initial performance measure estimates (each service center mean queue length estimate

non-negative and less than or equal to the actual value), superscripted by $v(0)$ and $v'(0)$, are such that, for each class s,

$$n_{ks}^{v(0)}(\vec{n}) \le n_{ks}^{v'(0)}(\vec{n}) \qquad 1 \le k \le K$$

$$n_{s\ pess}^{hv(0)}(\vec{n}) + \sum_{k=1}^{K} n_{ks}^{v(0)}(\vec{n}) \ge n_{s\ pess}^{hv'(0)}(\vec{n}) + \sum_{k=1}^{K} n_{ks}^{v'(0)}(\vec{n})$$

$$n_{s\ opt}^{hv(0)}(\vec{n}) + \sum_{k=1}^{K} n_{ks}^{v(0)}(\vec{n}) \le n_{s\ opt}^{hv'(0)}(\vec{n}) + \sum_{k=1}^{K} n_{ks}^{v'(0)}(\vec{n})$$

and, for each load-dependent service center k,

$$p_k^{v(0)}(m \mid \vec{n}) \le p_k^{v'(0)}(m \mid \vec{n}) \qquad 0 \le m \le c_k - 2$$

Then for $s \in C$, $i \ge 0$, and population $\vec{N}$ such that $|\vec{N}| = n+i$,

$$n_{ks}^{v(i)}(\vec{N}) \le n_{ks}^{v'(i)}(\vec{N}) \qquad 1 \le k \le K \tag{4.5-1}$$

$$n_{s\ pess}^{hv(i)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v(i)}(\vec{N}) \ge n_{s\ pess}^{hv'(i)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v'(i)}(\vec{N}) \tag{4.5-2}$$

$$n_{s\ opt}^{hv(i)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v(i)}(\vec{N}) \le n_{s\ opt}^{hv'(i)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v'(i)}(\vec{N}) \tag{4.5-3}$$

and, for each load-dependent service center k,

$$p_k^{v(i)}(m \mid \vec{N}) \le p_k^{v'(i)}(m \mid \vec{N}) \qquad 0 \le m \le c_k - 2 \tag{4.5-4}$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by applying the equations of the multiple closed class PBH algorithm, as modified to allow for load dependence, starting from values superscripted by $v(0)$ and $v'(0)$, respectively.

Proof:

The proof is by induction on i.

By assumption the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ ($i \ge 0$).

From the correctness of the modifications to the computation of $R_{s\ pess}^{(i)}(\vec{N})$, $a_s^{(i)}(\vec{N})$ and $b_s(\vec{N})$, and to the choices of $L_{sr\ UB}^{h}$ and $L_{sr\ LB}^{h}$, and from lemmas 4.1 to 4.3, relations (4.5-1) to (4.5-3) will be shown for $i+1$ if it can be shown that, for each load-dependent service center k and class $s \in C$,

$$R_{ks}^{v(i+1)}(\vec{N}) \le R_{ks}^{v'(i+1)}(\vec{N}) \tag{4.5-5}$$

From equation (4-28), relation (4.5-5) reduces to

$$\frac{L_{ks}(c_k)}{c_k}\left(1 + \sum_{r \in C} n_{kr}^{v(i)}(\vec{N} - \vec{e}_s)\right) + \sum_{m=1}^{c_k - 1} \left(L_{ks}(m) - L_{ks}(c_k)\frac{m}{c_k}\right) p_k^{v'(i)}(m-1 \mid \vec{N} - \vec{e}_s) \le$$

$$\frac{L_{ks}(c_k)}{c_k}\left(1 + \sum_{r \in C} n_{kr}^{v'(i)}(\vec{N} - \vec{e}_s)\right) + \sum_{m=1}^{c_k - 1} \left(L_{ks}(m) - L_{ks}(c_k)\frac{m}{c_k}\right) p_k^{v'(i)}(m-1 \mid \vec{N} - \vec{e}_s)$$

Since $L_{ks}(m) - L_{ks}(c_k)\frac{m}{c_k}$ is non-negative by assumption, this last relation holds from the inductive hypothesis.

From equation (4-29), relation (4.5-4) holds for $i+1$, $1 \le m \le c_k - 2$, if, for each class r,

$$L_{kr}(m)\ X_{r\ pess}^{v(i+1)}(\vec{N})\ p_k^{v(i)}(m-1 \mid \vec{N} - \vec{e}_r) \le L_{kr}(m)\ X_{r\ pess}^{v'(i+1)}(\vec{N})\ p_k^{v'(i)}(m-1 \mid \vec{N} - \vec{e}_r) \tag{4.5-6}$$

From relations (4.5-1) and (4.5-2), $X_{r\ pess}^{v(i+1)}(\vec{N}) \le X_{r\ pess}^{v'(i+1)}(\vec{N})$. In conjunction with

the inductive hypothesis, this establishes relation (4.5-6). For $m=0$, relation (4.5-4) becomes, from equation (4-30),

$$p_k^{\nu(i)}(0\,|\,\vec{N}-\vec{e}_r)\,\frac{X_{r\ pess}^{\nu(i+1)}(\vec{N})}{X_{r\ cpl}^{-k\ \nu(i+1)}(\vec{N})} \le p_k^{\nu'(i)}(0\,|\,\vec{N}-\vec{e}_r)\,\frac{X_{r\ pess}^{\nu'(i+1)}(\vec{N})}{X_{r\ opt}^{-k\ \nu'(i+1)}(\vec{N})} \qquad (4.5\text{-}7)$$

From relations (4.5-1) to (4.5-3), $X_{r\ pess}^{\nu(i+1)}(\vec{N}) \le X_{r\ pess}^{\nu'(i+1)}(\vec{N})$ and $X_{r\ opt}^{-k\ \nu(i+1)}(\vec{N}) \ge X_{r\ opt}^{-k\ \nu'(i+1)}(\vec{N})$. In conjunction with the inductive hypothesis, this implies relation (4.5-7). This establishes relation (4.5-4) and the lemma for $i+1$, and, by induction, for all $i \ge 0$. ∎

CHAPTER 5

LOOPED PERFORMANCE BOUND HIERARCHY ALGORITHMS

The accuracy of the PBH algorithm is dependent both on the form of the equations used, and on the accuracy of the initial performance measure estimates. The previous two chapters have focused on the algorithm's equations; the initial estimates that were used were chosen for simplicity as much as for accuracy. In this chapter, *Looped Performance Bound Hierarchy* (*LPBH*) algorithms are presented. These bounding algorithms use the PBH algorithm equations when generating higher level bounds; however, they also invest significant computational effort in obtaining the initial performance measure estimates.

The LPBH algorithms are so named due to the *looping algorithms* used in computing the initial performance measure estimates. The simplest type of looping algorithm for single class queueing networks, for example, uses an approximation giving the performance measures for the population with one fewer customer in terms of those for the original population. In conjunction with the MVA equations, which express the performance measures for the original population level in terms of those for the population with one fewer customer, this approximation allows an iteration, or looping, between the performance measures of adjacent population levels. Although the looping does not, in general, converge to the exact performance measure values, quite accurate approximations can be obtained. A major advantage of looping algorithms is that there is no additional cost incurred when the number of iterations to be performed is not chosen in advance, facilitating a goal of minimizing cost

under some given accuracy constraint. Note that in the PBH algorithm, this goal is more difficult to attain, since it is more expensive to compute bounds of multiple levels for a given population level than bounds of only a single level (see table 3.6).

Section one of this chapter treats the single class aMVA/PE algorithm. It is proved that the looping performed by this algorithm yields initial performance measure estimates suitable for the pessimistic computation of the single closed class PBH algorithm. Initial estimates suitable for the optimistic computation are provided by a looping algorithm given in section two, and are conjectured to be provided by a looping algorithm given in section three. Section four describes how a single closed class LPBH algorithm can be constructed using the looping algorithms given in the previous sections. Finally, section five presents a proposed multiple closed class LPBH algorithm. The chapter also has two associated appendices. Required lemmas are proved in appendix 5.A, while appendix 5.B contains results relevant to section one. The class of queueing networks being treated in this chapter are those with only fixed-rate and delay service centers, except when stated otherwise.

5.1. The Single Class aMVA/PE Algorithm*

5.1.1. Algorithm Definition

Although the aMVA/PE algorithm can be applied to multiple class queueing networks (as described in chapter two), it will be developed here for single class closed queueing networks only.

Viewed as a looping algorithm, the aMVA/PE algorithm is based on the following approximation giving population level $N-1$ performance measures in

* Portions of the material presented in this section appear in the paper "An Analysis of an Approximation Algorithm for Queueing Networks" [Eager & Sevcik 83b].

terms of those for population level N:

$$\tilde{R}_k(N-1) = R_k(N) \qquad 1 \le k \le K$$

and

$$\tilde{R}(N-1) = R(N)$$

From Little's equation,

$$\tilde{n}_k(N-1) = n_k(N)\frac{N-1}{N} \qquad 1 \le k \le K$$

The MVA equations are used to complete the looping, yielding the aMVA/PE equations

$$R_k^{(i)}(N) = L_k\left(1+n_k^{(i-1)}(N)\frac{N-1}{N}\right) \qquad 1 \le k \le K \tag{5-1}$$

$$R^{(i)}(N) = \sum_{k=1}^{K} R_k^{(i)}(N) \tag{5-2}$$

and

$$n_k^{(i)}(N) = \frac{R_k^{(i)}(N)}{Z+R^{(i)}(N)}N \qquad 1 \le k \le K \tag{5-3}$$

where the superscript (i) denotes values yielded by i iterations, or loopings, from initial performance measure estimates superscripted by (0). One suitable choice of initial estimates uses those for the pessimistic computation of the single closed class PBH algorithm (see subsection 3.1.2):

$$R^{(0)}(N) = b(N)$$

$$R_k^{(0)}(N) = \begin{cases} b(N) & \text{for } k = b \\ 0 & \text{for } k \ne b, 1 \le k \le K \end{cases}$$

and

$$n_k^{(0)}(N) = \begin{cases} \dfrac{b(N)}{Z+b(N)}N & \text{for } k = b \\ 0 & \text{for } k \ne b, 1 \le k \le K \end{cases}$$

In the next subsection, it is proved that the aMVA/PE equations have a unique solution (fixed point), and that the initial estimates given above guarantee monotonic convergence to this solution.

5.1.2. Monotonic Convergence to a Unique Solution

It is straightforward to prove that there is at most one solution to the aMVA/PE equations. This result depends on the assumption that the service center loadings are positive, and that only positive mean residence times are acceptable.

Theorem 5.1:

There exists at most one solution to the aMVA/PE equations.

Proof:

First of all, note that since a solution to the aMVA/PE equations is a fixed point of equations (5-1) to (5-3), at a solution the superscript (i) can be dropped.

For $N=1$, the aMVA/PE equations set $R_k(N) = L_k$ for all k, and the theorem holds. Consider the theorem for $N>1$. Substituting for $n_k(N)$ in equation (5-1), using equation (5-3), yields

$$R_k(N) = L_k\left[1+\frac{R_k(N)}{Z+R(N)}(N-1)\right] \qquad 1\le k\le K$$

which can be rewritten as

$$R_k(N)\left[1-\frac{(N-1)L_k}{Z+R(N)}\right] = L_k \tag{5-4}$$

Since L_k and $R_k(N)$ must be positive, this last equation implies that any acceptable solution must satisfy

$$R(N) > (N-1)L_k - Z \qquad 1\le k\le K \tag{5-5}$$

Isolating $R_k(N)$ on the left-hand side in equation (5-4), summing over k, and utilizing equation (5-2) yields

$$R(N) = \sum_{k=1}^{K}\frac{L_k}{1-\dfrac{(N-1)L_k}{Z+R(N)}}$$

$$<==> \quad 1 = \sum_{k=1}^{K}\left[\frac{L_k Z}{R(N)(R(N)-((N-1)L_k-Z))} + \frac{L_k}{R(N)-((N-1)L_k-Z)}\right] \tag{5-6}$$

In the region of interest as defined by inequality (5-5), and by the constraint $R(N) > 0$,

$$\sum_{k=1}^{K}\left[\frac{L_k Z}{R(N)(R(N)-((N-1)L_k-Z))} + \frac{L_k}{R(N)-((N-1)L_k-Z)}\right]$$

is a strictly decreasing function of $R(N)$. Therefore, the right-hand side of equation (5-6) can equal one for at most one value of $R(N)$, implying that there can be at most one acceptable value of $R(N)$ satisfying equation (5-6), and, equivalently, the aMVA/PE equations. Since the $R_k(N)$ values, and therefore the $n_k(N)$ values, are uniquely determined by $R(N)$ (equation (5-4)), the theorem is established. ∎

The next theorem shows that the initial performance measure estimates of the previous subsection guarantee convergence to a (necessarily unique) solution. This convergence is monotonic in the sense that each of the performance measure values produced at the i+1st iteration lies in the interval defined by that produced at the ith iteration and that of the exact solution to the equations.

Theorem 5.2:

The initial performance measure estimates of subsection 5.1.1 guarantee monotonic convergence to a solution of the aMVA/PE equations.

Proof:

It is sufficient to establish that the mean queue length estimates produced after i+1 iterations are bottleneck majorized by those produced after i iterations, for all $i \geq 0$. This shows, in particular, that $R^{(i+1)}(N) \leq R^{(i)}(N)$, which is sufficient to imply convergence since $R^{(i)}(N)$ is bounded below by 1 for all $i \geq 0$.

Consider the mean queue length estimates that result after one iteration:

$$n_k^{(1)}(N) = \begin{cases} \dfrac{L_b\left(1+\dfrac{b(N)}{Z+b(N)}(N-1)\right)}{1+\dfrac{b(N)}{Z+b(N)}(N-1)L_b} & \text{for } k=b \\ \dfrac{L_k}{1+\dfrac{b(N)}{Z+b(N)}(N-1)L_b} & \text{for } k\neq b, 1\leq k\leq K \end{cases}$$

It is easily verified that these estimates are bottleneck majorized by the initial mean queue length estimates, or that

$$n_b^{(1)}(N) \leq n_b^{(0)}(N)$$
$$n_k^{(1)}(N) \geq n_k^{(0)}(N) \qquad k\neq b, 1\leq k\leq K$$

and

$$\sum_{k=1}^{K} n_k^{(0)}(N) \geq \sum_{k=1}^{K} n_k^{(1)}(N)$$

By lemma 5.1 in appendix 5.A, the aMVA/PE equations preserve bottleneck majorization. This establishes the theorem. ∎

Many other initial estimate choices can also be shown to yield monotonic convergence to the solution of the aMVA/PE equations. In appendix 5.B, an example is given in which the estimates correspond to the optimistic assumption that the mean residence time at each service center is equal to the service center loading. In this case, however, the convergence is from the other side of the exact solution to the equations (the optimistic side). Agrawal has shown convergence, for queueing networks with no delay service centers, when the initial mean queue length estimate for each service center is $\frac{N}{K}$ [Agrawal 83].

5.1.3. The Performance Measures Produced

In theorem 5.3, it is shown that the level i mean queue lengths from the pessimistic computation of the single closed class PBH algorithm are majorized by the mean queue lengths yielded after i iterations using the aMVA/PE equations, starting from the initial performance measure estimates given in

subsection 5.1.1. This implies that the pessimistic PBH performance measures are more accurate than the corresponding performance measures yielded by the aMVA/PE algorithm. Also, since majorization is transitive and the exact mean queue lengths are majorized by the level i PBH mean queue lengths, this result implies that the exact mean queue lengths are majorized by the aMVA/PE mean queue lengths, implying that the aMVA/PE performance measures are suitable as initial performance measure estimates for the pessimistic computation of the PBH algorithm. (Although the initial estimates given previously for the pessimistic computation are based on bottleneck majorization, majorization is sufficient, as will be shown in section 5.4.) In the theorem, the performance measures from the pessimistic computation of the PBH algorithm are distinguished by the subscript *pess*.

Theorem 5.3:

Consider an indexing of the fixed-rate service centers such that $L_k \geq L_{k+1}$, $1 \leq k \leq K-1$. For all $i \geq 0$, $N \geq 1$,

$$n_j^{(i)}{}_{pess}(N) \geq n_j^{(i)}(N) \frac{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N)}{\sum_{k=1}^{j} n_k^{(i)}(N)} \qquad 1 \leq j \leq K \tag{5-7}$$

and

$$\sum_{k=1}^{K} n_k^{(i)}(N) \geq \sum_{k=1}^{K} n_k^{(i)}{}_{pess}(N) \tag{5-8}$$

Proof:

The proof is by induction on i.

Since $n_k^{(0)}{}_{pess}(N) = n_k^{(0)}(N)$ for all k, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ ($i \geq 0$). From lemma 5.2 in appendix 5.A,

$$n_j^{(i)}{}_{pess}(N-1) \geq n_j^{(i)}{}_{pess}(N) \frac{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N-1)}{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N)} \qquad 1 \leq j \leq K \tag{5-9}$$

and

$$\frac{N-1}{N} \sum_{k=1}^{K} n_k^{(i)}{}_{pess}(N) \geq \sum_{k=1}^{K} n_k^{(i)}{}_{pess}(N-1)$$

Applying the inductive hypothesis to this last relation yields

$$\sum_{k=1}^{K} n_k^{(i)}(N) \frac{N-1}{N} \geq \sum_{k=1}^{K} n_k^{(i)}{}_{pess}(N-1) \tag{5-10}$$

Applying the inductive hypothesis to relation (5-9) yields

$$n_j^{(i)}{}_{pess}(N-1) \geq n_j^{(i)}(N) \frac{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N-1)}{\sum_{k=1}^{j} n_k^{(i)}(N)} \qquad 1 \leq j \leq K$$

implying that

$$n_j^{(i)}{}_{pess}(N-1) \geq n_j^{(i)}(N) \frac{N-1}{N} \frac{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N-1)}{\sum_{k=1}^{j} n_k^{(i)}(N) \frac{N-1}{N}} \qquad 1 \leq j \leq K \tag{5-11}$$

Using the PBH and aMVA/PE equations, for $i+1$ and $1 \leq j \leq K$ relation (5-7) reduces to (after canceling common factors from both sides of the relation)

$$1+n_j^{(i)}{}_{pess}(N-1) \geq (1+n_j^{(i)}(N) \frac{N-1}{N}) \frac{\sum_{k=1}^{j} L_k (1+n_k^{(i)}{}_{pess}(N-1))}{\sum_{k=1}^{j} L_k (1+n_k^{(i)}(N) \frac{N-1}{N})}$$

This relation holds from lemma 3.2 applied to relations (5-10) and (5-11) (where n_k is taken as $n_k^{(i)}(N) \frac{N-1}{N}$, and $\hat{n}_k$ is taken as $n_k^{(i)}{}_{pess}(N-1)$, for $1 \leq k \leq K$), establishing relation (5-7) for $i+1$.

Relation (5-8) for $i+1$ reduces to

$$\frac{\sum_{k=1}^{K} L_k (1+n_k^{(i)}(N) \frac{N-1}{N})}{Z+\sum_{k=1}^{K} L_k (1+n_k^{(i)}(N) \frac{N-1}{N})} N \geq \frac{\sum_{k=1}^{K} L_k (1+n_k^{(i)}{}_{pess}(N))}{Z+\sum_{k=1}^{K} L_k (1+n_k^{(i)}{}_{pess}(N))} N$$

$$<==> \quad \sum_{k=1}^{K} L_k(1+n_k^{(i)}(N)\frac{N-1}{N}) \geq \sum_{k=1}^{K} L_k(1+n_{k\,pess}^{(i)}(N))$$

This last relation holds from lemma 3.2 applied to relations (5-10) and (5-11) as before, establishing relation (5-8) and the theorem for $i+1$. By induction, the theorem is established for $i \geq 0$. ∎

5.1.4. An Example

Table 5.1 gives performance measures yielded by the aMVA/PE algorithm for the queueing network of table 3.1, for a population of 50 customers. Both the performance measures obtained after 5 iterations, and the performance measures at convergence (which required about 16 iterations for the precision used in the table) are shown. The exact performance measures are given for comparison.

Another comparison that can be made is between the performance measures produced after 5 iterations, and the level 5 performance measures of the pessimistic computation of the single closed class PBH algorithm. As predicted by theorem 5.3, the latter performance measures are more accurate. For example, the mean system residence time value yielded by the PBH pessimistic computation is 2.33 (table 3.2), as contrasted to 2.44 as yielded by the aMVA/PE algorithm. The relationship between the performance measures yielded by a looping algorithm such as the aMVA/PE algorithm, and those yielded by the PBH algorithm, is discussed further in subsection 5.4.1.

5.2. A Single Class Optimistic Looping Algorithm

5.2.1. Algorithm Definition

This section describes a looping algorithm that yields performance measures suitable as initial estimates for the optimistic computation of the single closed class PBH algorithm. The looping algorithm is based on the following

System Performance Measures					
System Throughput			Mean System Residence Time		
5 Iter.	Conv.	Exact	5 Iter.	Conv.	Exact
2.87	2.90	2.92	2.44	2.22	2.15

Fixed-Rate Service Center Performance Measures								
Service Center	Mean Residence Time			Mean Queue Length			Utilization		
	5 Iter.	Conv.	Exact	5 Iter.	Conv.	Exact	5 Iter.	Conv.	Exact
1	1.15	.87	.81	3.31	2.53	2.35	.72	.73	.73
2	.43	.46	.46	1.22	1.35	1.33	.57	.58	.58
3	.31	.33	.33	.90	.96	.95	.49	.49	.50
4	.28	.29	.29	.81	.85	.85	.46	.46	.47
5	.10	.10	.10	.29	.30	.30	.23	.23	.23
6	.07	.07	.07	.21	.21	.21	.17	.17	.17
7	.06	.06	.06	.17	.17	.17	.14	.15	.15
8	.03	.03	.03	.09	.10	.09	.09	.09	.09

Delay Service Center Performance Measures		
Mean Queue Length		
5 Iter.	Conv.	Exact
43.00	43.54	43.74

aMVA/PE Performance Measures for an Example Queueing Network

Table 5.1

approximation giving population level $N-1$ service center mean residence times in terms of those for population level N:

$$\tilde{R}_k(N-1) = R_k(N) - \frac{L_k^N}{L^{(N-1)}}\Delta_{UB}(N) \qquad 1 \le k \le K$$

where $\Delta(N)$ is the change in the mean number of customers at the fixed-rate service centers when going from a population level of $N-1$ to a population level of N, and $\Delta_{UB}(N)$ is an upper bound on this quantity defined by

$$\Delta_{UB}(N) = \frac{R(N)}{Z+R(N)}N - \frac{R(N)-L_b}{Z+R(N)-L_b}(N-1)$$

The fact that $\Delta_{UB}(N)$ is indeed an upper bound follows from the relation $R(N-1) \geq R(N)-L_b$, which can be proved, in conjunction with $n_k(N) \geq n_k(N-1)$ for all fixed-rate service centers k, by induction on N.

Using $R(N-1) = \sum_{k=1}^{K} R_k(N-1)$ yields the following approximation for $R(N-1)$:

$$\tilde{R}(N-1) = R(N) - \frac{L^{(N)}}{L^{(N-1)}}\Delta_{UB}(N)$$

Note that $\frac{1}{K} \leq \frac{L^{(N)}}{L^{(N-1)}} \leq L_b$ (as follows from $L^{(p)}=\sum_{k=1}^{K} L_k^p$ for all p). Applying Little's equation yields

$$\tilde{n}_k(N-1) = \frac{R_k(N) - \frac{L_k^N}{L^{(N-1)}}\Delta_{UB}(N)}{Z+R(N) - \frac{L^{(N)}}{L^{(N-1)}}\Delta_{UB}(N)}(N-1) \qquad 1 \leq k \leq K$$

The MVA equations are used to complete the looping, yielding the following equations for the optimistic looping algorithm:

$$R_k^{(i)}(N) = L_k(1+\tilde{n}_k^{(i-1)}(N-1)) \qquad 1 \leq k \leq K \tag{5-12}$$

$$R^{(i)}(N) = \sum_{k=1}^{K} R_k^{(i)}(N) \tag{5-13}$$

$$\tilde{n}_k^{(i)}(N-1) = \frac{R_k^{(i)}(N) - \frac{L_k^N}{L^{(N-1)}}\Delta_{UB}^{(i)}(N)}{Z+R^{(i)}(N) - \frac{L^{(N)}}{L^{(N-1)}}\Delta_{UB}^{(i)}(N)}(N-1) \qquad 1 \leq k \leq K \tag{5-14}$$

and

$$\Delta_{UB}^{(i)}(N) = \frac{R^{(i)}(N)}{Z+R^{(i)}(N)}N - \frac{R^{(i)}(N)-L_b}{Z+R^{(i)}(N)-L_b}(N-1) \tag{5-15}$$

where the superscript (i) denotes values produced by i iterations from initial performance measure estimates superscripted by (0). One suitable choice of initial estimates uses the level 1 mean residence times from the optimistic computation of the basic PBH algorithm (the level 0 mean residence times are not used since they may result in negative $\tilde{n}_k^{(1)}(N-1)$ values):

$$R^{(0)}(N) = 1+\frac{1}{K}\left(\frac{a(N-1)}{Z+a(N-1)}\right)(N-1)$$

and

$$R_k^{(0)}(N) = L_k\left(1+\frac{1}{K}\left(\frac{a(N-1)}{Z+a(N-1)}\right)\right)(N-1) \qquad 1\le k\le K$$

For $1\le k\le K$, $\tilde{n}_k^{(0)}(N-1)$ is computed using equation (5-14).

It is unknown as to whether the optimistic looping algorithm equations have a unique solution. It has also not been established that the initial estimates given above guarantee monotonic convergence to a solution, although one can prove that these equations preserve majorization (which implies that if the initial estimates are majorized by the estimates yielded after one iteration, then convergence will be monotonic, in the sense that the performance measures produced will become less and less optimistic.)

5.2.2. The Performance Measures Produced

It is shown in theorem 5.4 that the performance measures yielded by the optimistic looping algorithm are suitable as initial performance measure estimates for the optimistic computation of the single closed class PBH algorithm, by showing that the service center mean queue lengths estimates produced are majorized by the exact mean queue lengths.

Theorem 5.4: Consider an indexing of the fixed-rate service centers such that $L_k \ge L_{k+1}$, $1\le k\le K-1$. For all $i\ge 0$, $N\ge 1$,

$$n_j^{(i)}(N) \ge n_j(N)\frac{\sum_{k=1}^{j} n_k^{(i)}(N)}{\sum_{k=1}^{j} n_k(N)} \qquad 1\le j\le K \tag{5-16}$$

and

$$\sum_{k=1}^{K} n_k(N) \ge \sum_{k=1}^{K} n_k^{(i)}(N) \tag{5-17}$$

Proof:

The proof is by induction on i.

For $i=0$, the theorem holds from the choice of initial performance measure estimates, since the optimistic PBH mean queue lengths are majorized by the exact mean queue lengths (theorem 3.1). This establishes the induction basis.

Assume that the theorem holds for i ($i \geq 0$), and consider the theorem for $i+1$. For $N=1$, relations (5-16) and (5-17) hold with equality; therefore, only $N \geq 2$ need be considered. It will first be established that

$$\tilde{n}_j^{(i)}(N-1) \geq \tilde{n}_j(N-1) \frac{\sum_{k=1}^{j} \tilde{n}_k^{(i)}(N-1)}{\sum_{k=1}^{j} \tilde{n}_k(N-1)} \qquad 1 \leq j \leq K \tag{5-18}$$

and

$$\sum_{k=1}^{K} \tilde{n}_k(N-1) \geq \sum_{k=1}^{K} \tilde{n}_k^{(i)}(N-1) \tag{5-19}$$

where $\tilde{n}_k^{(i)}(N-1)$ is defined by equation (5-14), and $\tilde{n}_k(N-1)$ is defined by

$$\tilde{n}_k(N-1) = \frac{R_k(N) - \frac{L_k^N}{L^{(N-1)}}\Delta(N-1)}{Z+R(N) - \frac{L^{(N)}}{L^{(N-1)}}\Delta(N-1)}(N-1) \qquad 1 \leq k \leq K \tag{5-20}$$

Relation (5-19) is equivalent to

$$R(N) - \frac{L^{(N)}}{L^{(N-1)}}\Delta(N-1) \geq R^{(i)}(N) - \frac{L^{(N)}}{L^{(N-1)}}\Delta_{JB}^{(i)}(N)$$

It is a consequence of the concavity of throughput (theorem 5.5 in section 5.3) that $\Delta(N-1) \leq \Delta(N)$. Also, as noted previously, $\Delta(N) \leq \Delta_{JB}(N)$. Therefore, this last relation will be shown if it can be shown with $\Delta(N-1)$ replaced by $\Delta_{JB}(N)$. Since $R(N) \geq R^{(i)}(N)$ (from $\sum_{k=1}^{K} n_k(N) \geq \sum_{k=1}^{K} n_k^{(i)}(N)$ as given by the inductive hypothesis), the relation trivially holds if $\Delta_{JB}^{(i)}(N) \geq \Delta_{JB}(N)$. Suppose that $\Delta_{JB}^{(i)}(N) < \Delta_{JB}(N)$. Then, since $R^{(i)}(N) \geq 1 \geq \frac{L^{(N)}}{L^{(N-1)}}$, the relation holds if

$$R(N) - R^{(i)}(N)(R(N)\frac{\Delta_{VB}(N)}{R(N)}) \geq R^{(i)}(N) - R^{(i)}(N)(R^{(i)}(N)\frac{\Delta^{(j)}_{VB}(N)}{R^{(i)}(N)})$$

$$<==> \quad R(N)(1-R^{(i)}(N)\frac{\Delta_{VB}(N)}{R(N)}) \geq R^{(i)}(N)(1-R^{(i)}(N)\frac{\Delta^{(j)}_{VB}(N)}{R^{(i)}(N)})$$

which follows from $R(N) \geq R^{(i)}(N)$, and from $\frac{\Delta_{VB}(N)}{R(N)} \leq \frac{\Delta^{(j)}_{VB}(N)}{R^{(i)}(N)}$ (This last relation can be established using $R(N) \geq R^{(i)}(N)$.)

Using equations (5-14) and (5-20), for $1 \leq j \leq K$ relation (5-18) reduces to

$$R^{(i)}_j(N) - \frac{L^N_j}{L^{(N-1)}}\Delta^{(j)}_{VB}(N) \geq$$

$$(R_j(N) - \frac{L^N_j}{L^{(N-1)}}\Delta(N-1))\frac{\sum_{k=1}^{j}(R^{(i)}_k(N) - \frac{L^N_k}{L^{(N-1)}}\Delta^{(j)}_{VB}(N))}{\sum_{k=1}^{j}(R_k(N) - \frac{L^N_k}{L^{(N-1)}}\Delta(N-1))} \qquad (5\text{-}21)$$

Since

$$R^{(i)}_j(N) \leq R_j(N)\frac{\sum_{k=1}^{j}R^{(i)}_k(N)}{\sum_{k=1}^{j}R_k(N)}$$

from the inductive hypothesis on relation (5-16) and from Little's equation, relation (5-21) will be shown if it can be shown when $R^{(i)}_j(N)$ is replaced by $R_j(N)\frac{\sum_{k=1}^{j}R^{(i)}_k(N)}{\sum_{k=1}^{j}R_k(N)}$ on the left-hand side of this relation. Making this substitution and then simplifying yields

$$R_j(N)(\sum_{k=1}^{j}L^N_k)(\Delta^{(j)}_{VB}(N)\sum_{k=1}^{j}R_k(N) - \Delta(N-1)\sum_{k=1}^{j}R^{(i)}_k(N)) \geq$$

$$L^N_j(\sum_{k=1}^{j}R_k(N))(\Delta^{(j)}_{VB}(N)\sum_{k=1}^{j}R_k(N) - \Delta(N-1)\sum_{k=1}^{j}R^{(i)}_k(N))$$

Since $R_j(N)\sum_{k=1}^{j}L^N_k \geq L^N_j\sum_{k=1}^{j}R_k(N)$ from lemma 5.5 in appendix 5.A, and $\Delta(N-1) \leq \Delta_{VB}(N)$, this last relation will be shown if it can be shown that

$$\Delta^{(j)}_{VB}(N)\sum_{k=1}^{j}R_k(N) \geq \Delta_{VB}(N)\sum_{k=1}^{j}R^{(i)}_k(N)$$

$$<==> \quad \frac{\Delta_{UB}^{(i)}(N)}{R^{(i)}(N)} \sum_{k=1}^{j} n_k(N) \geq \frac{\Delta_{UB}(N)}{R(N)} \sum_{k=1}^{j} n_k^{(i)}(N) \frac{\sum_{k=1}^{K} n_k(N)}{\sum_{k=1}^{K} n_k^{(i)}(N)}$$

This last relation holds from lemma 5.6 in appendix 5.A applied to the inductive hypothesis, and from $\frac{\Delta_{UB}^{(i)}(N)}{R^{(i)}(N)} \geq \frac{\Delta_{UB}(N)}{R(N)}$. This establishes relation (5-18).

Since majorization is transitive here, lemma 5.7 in appendix 5.A and relations (5-18) and (5-19) imply that

$$\tilde{n}_j^{(i)}(N-1) \geq n_j(N-1) \frac{\sum_{k=1}^{j} \tilde{n}_k^{(i)}(N-1)}{\sum_{k=1}^{j} n_k(N-1)}$$

and

$$\sum_{k=1}^{K} n_k(N-1) \geq \sum_{k=1}^{K} \tilde{n}_k^{(i)}(N-1)$$

Applying lemma 3.2 to these last two relations (with $\tilde{n}_k$ taken as $\tilde{n}_k^{(i)}(N-1)$ and n_k taken as $n_k(N-1)$, $1 \leq k \leq K$) yields relation (5-17) for $i+1$ (noting that relation (5-17) for $i+1$ is equivalent to $\sum_{k=1}^{K} L_k(1+n_k(N-1)) \geq \sum_{k=1}^{K} L_k(1+\tilde{n}_k^{(i)}(N-1))$). Using relations (5-14) and (5-20), for $1 \leq j \leq K$ relation (5-16) reduces to (after canceling common factors from both sides of the relation)

$$1+\tilde{n}_j^{(i)}(N-1) \geq (1+n_j(N-1)) \frac{\sum_{k=1}^{j} L_k(1+\tilde{n}_k^{(i)}(N-1))}{\sum_{k=1}^{j} L_k(1+n_k(N-1))}$$

which holds from lemma 3.2 applied as before. This establishes the theorem for $i+1$, and, by induction, for all $i \geq 0$. ∎

5.2.3. An Example

Table 5.2 gives performance measures yielded by the optimistic looping algorithm for the queueing network of table 3.1, for a population of 50 custo-

mers. Both the performance measures obtained after 4 iterations, and the performance measures at convergence (which required about 8 iterations for the precision used in the table) are shown. The exact performance measures are given for comparison.

System Performance Measures					
System Throughput			Mean System Residence Time		
4 Iter.	Conv.	Exact	4 Iter.	Conv.	Exact
2.98	2.97	2.92	1.77	1.81	2.15

Fixed-Rate Service Center Performance Measures								
Service Center	Mean Residence Time			Mean Queue Length			Utilization		
	4 Iter.	Conv.	Exact	4 Iter.	Conv.	Exact	4 Iter.	Conv.	Exact
1	.39	.41	.81	1.18	1.22	2.35	.75	.74	.73
2	.47	.49	.46	1.39	1.45	1.33	.60	.59	.58
3	.34	.34	.33	1.00	1.01	.95	.51	.51	.50
4	.30	.30	.29	.90	.90	.85	.48	.48	.47
5	.10	.10	.10	.31	.31	.30	.24	.24	.23
6	.07	.07	.07	.22	.22	.21	.18	.18	.17
7	.06	.06	.06	.18	.17	.17	.15	.15	.15
8	.03	.03	.03	.10	.10	.09	.09	.09	.09

Delay Service Center Performance Measures		
Mean Queue Length		
4 Iter.	Conv.	Exact
44.73	44.62	43.74

Optimistic Looping Algorithm Performance Measures for an Example Queueing Network

Table 5.2

Another comparison that can be made is between the performance measures produced after 4 iterations, and the level 5 performance measures of the optimistic computation of the basic single closed class PBH algorithm. (Recall that the level 0 performance measures of the optimistic looping algorithm

correspond to the level 1 performance measures of the PBH algorithm.) In this example, and, empirically, in general, the latter performance measures are more accurate. For example, the mean system residence time value yielded by the PBH optimistic computation is 2.07 (table 3.2), as contrasted to 1.77 as yielded by the optimistic looping algorithm. This comparison is discussed further in subsection 5.4.1.

5.3. A Conjectured Single Class Optimistic Looping Algorithm*

5.3.1. Motivation

The optimistic looping algorithm of the previous section is, empirically, less accurate than the pessimistic looping algorithm of section 5.1 (the single class aMVA/PE algorithm). In this section, an alternative optimistic looping algorithm is presented. This algorithm is the optimistic analogue of the pessimistic looping algorithm, and empirically can yield performance measures that are much more accurate (but also more costly) than those of the looping algorithm of the previous section. However, it is only a conjecture that optimistic performance measure estimates are produced.

The pessimistic nature of the looping algorithm of section 5.1 is, intuitively, due to the fact that it is pessimistic (with respect to the performance measures of population level $N-1$) to assume that the mean system and service center residence times with $N-1$ customers are the same as those with N customers. For the mean system residence time, this is a consequence of the fact that it is an increasing function of the queueing network population level.

Although apparently it has not been proved previously, it is also the case that mean system residence time is an upwards convex function of the queue-

* Portions of the material presented in this section appear in the note "A proof of throughput concavity and residence time convexity" [Eager 83].

ing network population level. More generally, it is optimistic (with respect to the performance measures of population level $N-1$) to assume that the changes in the mean system and service center residence times in going from N to $N-1$ customers are the same as those in going from $N+1$ to N customers. This is proved in lemma 5.9 in appendix 5.A by showing that the population level $N-1$ mean queue length estimates derived with the use of this assumption are majorized by the exact mean queue lengths. Theorem 5.5 gives the corresponding results for the system performance measures of mean system residence time and system throughput. The result for system throughput has been proved independently by Dowdy, Gordon, and Saxton [Dowdy, Gordon & Saxton 83]; our results are jointly reported in [Dowdy et al. 83].

Theorem 5.5:

For all $N \geq 2$,

$$R(N) - R(N-1) \leq R(N+1) - R(N) \qquad (5\text{-}21)$$

$$X(N) - X(N-1) \geq X(N+1) - X(N) \qquad (5\text{-}22)$$

Proof:

Relation (5-21) is a direct consequence of relation (5.9-2) in lemma 5.9 (it also appears in the lemma as relation (5.9-5)).

Relation (5-22) establishes the concavity of system throughput, and will be proved here with the use of relation (5-21). Using Little's equation, relation (5-22) is equivalent to

$$\frac{N}{Z+R(N)} - \frac{N-1}{Z+R(N-1)} \geq \frac{N+1}{Z+R(N+1)} - \frac{N}{Z+R(N)}$$

This relation holds if it holds for $R(N-1)$ minimized, which by relation (5-21) corresponds to $R(N-1) = 2R(N)-R(N+1)$. Making this substitution and simplifying results in

$$(Z+R(N))(R(N+1)-R(N)) \geq N(R(N+1)-R(N))^2$$

Since $R(N+1) \geq R(N)$, this reduces to

$$Z+R(N) \geq N(R(N+1)-R(N))$$
$$<==> \quad (N+1)(R(N)+Z) \geq N(R(N+1)+Z)$$

This last relation holds from Little's equation, and the fact that $X(N+1) \geq X(N)$. This completes the proof of the theorem. ∎

The conjectured optimistic looping algorithm to be presented here is based on the fact that mean system residence time is convex (and on the corresponding service center mean residence time properties given by lemma 5.9) much as the pessimistic looping algorithm of section 5.1 is based on the fact that mean system residence time is increasing (and on the service center mean residence time properties given by lemma 5.2 or 5.3).

5.3.2. Algorithm Structure

The conjectured optimistic looping algorithm uses the following approximation giving population level $N-1$ service center mean residence times in terms of those for population levels N and $N+1$:

$$\tilde{R}_k(N-1) = R_k(N) - (R_k(N+1) - R_k(N)) \qquad 1 \leq k \leq K$$

Using $R(N-1) = \sum_{k=1}^{K} R_k(N-1)$ yields the following approximation for $R(N-1)$:

$$\tilde{R}(N-1) = R(N) - (R(N+1) - R(N))$$

The MVA equations are used to complete the looping, yielding the following equations for the conjectured optimistic looping algorithm ($i \geq 1$):

$$R_k^{(i)}(N) = L_k(1+\tilde{n}_k^{(i-1)}(N-1)) \qquad 1 \leq k \leq K \tag{5-23}$$

$$R^{(i)}(N) = \sum_{k=1}^{K} R_k^{(i)}(N) \tag{5-24}$$

$$\tilde{n}_k^{(i)}(N-1) = \frac{R_k^{(i)}(N)-(R_k^{(i)}(N+1)-R_k^{(i)}(N))}{Z+R^{(i)}(N)-(R^{(i)}(N+1)-R^{(i)}(N))}(N-1) \qquad 1 \leq k \leq K \tag{5-25}$$

$$R_k^{(i)}(N+1) = L_k(1+n_k^{(i-1)}(N)) \qquad 1 \leq k \leq K \tag{5-26}$$

$$R^{(i)}(N+1) = \sum_{k=1}^{K} R_k^{(i)}(N+1) \tag{5-27}$$

and

$$n_k^{(i)}(N) = \frac{R_k^{(i)}(N)}{Z+R^{(i)}(N)}N \qquad 1 \le k \le K \tag{5-28}$$

where the subscript (i) denotes values produced by i iterations from initial performance measure estimates superscripted by (0). One suitable choice of initial estimates uses those for the optimistic computation of the PBH algorithm:

$$R^{(0)}(N) = a(N)$$
$$R_k^{(0)}(N) = \frac{a(N)}{K} \qquad 1 \le k \le K$$

and

$$n_k^{(0)}(N) = \frac{1}{K}\frac{a(N)}{Z+a(N)}N \qquad 1 \le k \le K$$

These initial estimates must be calculated for population levels N and $N+1$. Equation (5-25) is then applied to obtain the $\tilde{n}_k^{(0)}(N-1)$ values.

The looping used in equations (5-23) to (5-28) is more complex than that used in the previous looping algorithms. Each iteration computes new performance measure estimates for population levels N and $N+1$. The new estimates for population level $N+1$ are calculated from the old estimates for population level N using the MVA equations. The new estimates for population level N are obtained by applying the MVA equations to performance measure estimates for population level $N-1$, which are calculated using the assumption that the changes in mean residence times in going from N to $N-1$ customers are the same as those in going from $N+1$ to N customers.

Note that this looping requires two applications of the MVA equations in each iteration. Therefore, an iteration of the conjectured optimistic looping algorithm is about twice as expensive as an iteration of one of the previous looping algorithms.

Although empirically the looping algorithm presented here yields optimistic performance measure estimates, this has not been proved.

Conjecture:

The population level N mean queue length estimates obtained by iterating using equations (5-23) to (5-28), from the given initial performance measure estimates, are majorized by the exact mean queue lengths.

Discussion:

The major analytic result supporting this conjecture is that the use of the exact performance measures as initializations yields, after one iteration, population level N mean queue length estimates that are majorized by the exact mean queue lengths. This result is a consequence of lemma 5.9, which implies that the approximate population level $N-1$ mean queue lengths produced from such initial estimates are majorized by the exact mean queue lengths, and of the fact that the MVA equations preserve majorization.

5.3.3. An Example

Table 5.3 gives performance measures yielded by the conjectured optimistic looping algorithm for the queueing network of table 3.1, for a population of 50 customers. Both the performance measures obtained after 5 iterations, and the performance measures at convergence (which required about 10 iterations for the precision used in the table) are shown. The exact performance measures are given for comparison.

Note that the performance measures produced by the conjectured optimistic looping algorithm are more accurate than those produced by the optimistic looping algorithm of the previous section (table 5.2). Another comparison that can be made is between the performance measures produced after 5 iterations, and the level 5 performance measures of the optimistic computation of the basic single closed class PBH algorithm. In this example, and, empirically, in general, the latter performance measures are only marginally more accurate.

System Performance Measures					
System Throughput			Mean System Residence Time		
5 Iter.	Conv.	Exact	5 Iter.	Conv.	Exact
2.93	2.92	2.92	2.07	2.14	2.15

Fixed-Rate Service Center Performance Measures								
Service Center	Mean Residence Time			Mean Queue Length			Utilization	
	5 Iter.	Conv.	Exact	5 Iter.	Conv.	Exact	5 Iter.	Conv.	Exact
1	.74	.80	.81	2.15	2.33	2.35	.73	.73	.73
2	.45	.45	.46	1.31	1.33	1.33	.59	.58	.58
3	.33	.33	.33	.95	.95	.95	.50	.50	.50
4	.29	.29	.29	.86	.85	.85	.47	.47	.47
5	.10	.10	.10	.30	.30	.30	.23	.23	.23
6	.07	.07	.07	.21	.21	.21	.18	.18	.17
7	.06	.06	.06	.17	.17	.17	.15	.15	.15
8	.03	.03	.03	.10	.10	.09	.09	.09	.09

Delay Service Center Performance Measures		
Mean Queue Length		
5 Iter.	Conv.	Exact
43.94	43.76	43.74

Conjectured Optimistic Looping Algorithm Performance Measures for an Example Queueing Network

Table 5.3

For example, the mean system residence time value yielded by the optimistic looping algorithm is the same (to the precision shown in the table) as that yielded by the optimistic PBH computation.

5.4. A Looped Single Closed Class Performance Bound Hierarchy Algorithm

5.4.1. Motivation

A LPBH algorithm uses the looping algorithms of the previous sections when obtaining initial performance measure estimates. However, the single closed class PBH algorithm itself could be applied to yield more accurate initial

estimates at the same or at a lower cost. This is illustrated in table 5.4, which collects results from tables 3.2, 5.1, 5.2 and 5.3. The PBH bounds have the same or a lower cost than those of the looping algorithms (see subsection 5.4.5).

System Throughput					
Pessimistic Bounds		Exact	Optimistic Bounds		
Looping (5 iter.)	PBH (Level 5)		PBH (Level 5)	Conj. Looping (5 iter.)	Looping (4 iter.)
2.87	2.89	2.92	2.93	2.93	2.98

Mean System Residence Time					
Pessimistic Bounds		Exact	Optimistic Bounds		
Looping (5 iter.)	PBH (Level 5)		PBH (Level 5)	Conj. Looping (5 iter.)	Looping (4 iter.)
2.44	2.33	2.15	2.07	2.07	1.77

A Comparison Between Looping Algorithms and the PBH Algorithm

Table 5.4

An advantage of looping algorithms, which compensates for their lesser accuracy and motivates their development, is that there is no computational cost penalty in not choosing in advance the number of iterations to be performed. This allows the number of iterations to be chosen dynamically, based upon the tightness of the bounds produced at each iteration. With the PBH algorithm, especially the non-symbolic formulation, the bound level must be chosen in advance if the computational cost penalty incurred in computing bounds of multiple levels is to be avoided.

5.4.2. Algorithm Definition

A LPBH algorithm can be constructed using the pessimistic looping algo-

rithm of section 5.1, and either the optimistic looping algorithm of section 5.2, or that of section 5.3. Since it is only a conjecture that the latter algorithm yields optimistic performance measure estimates, the optimistic looping algorithm of section 5.2 will be used here.

The resulting single closed class LPBH algorithm, when applied to queueing networks with no open classes, has two computational phases. In the first phase, initial performance measure estimates for population level $N-i$, where i is the chosen bound level, are computed using the pessimistic and optimistic looping algorithms. In the second phase, both pessimistic and optimistic performance measure estimates are calculated for successively higher population levels, until population level N is attained, starting from the pessimistic and optimistic, respectively, initial estimates for population level $N-i$. The equations used in the second phase may be those of either the modified or the basic single closed class PBH algorithm; for clarity, the equations of the basic PBH algorithm will be used here.

The first phase provides a tradeoff between computational effort and accuracy, since any number of iterations may be used in the two looping algorithms. Although the number of iterations need not be the same, for clarity attention will be restricted to the case in which there is a common looping algorithm *iteration level*. With each looping algorithm, iterations are performed until either convergence occurs, or the iteration level is attained. To facilitate comparison with the PBH algorithm, an iteration level of l allows l iterations in the pessimistic looping algorithm, but only $l-1$ iterations in the optimistic looping algorithm.

The looping algorithm iteration level can be chosen dynamically, depending on the tightness of the initial performance measure estimates being produced. The bound level, which provides a tradeoff between computational effort

and accuracy in the second computational phase, can also be chosen dynamically, but it is computationally efficient to do so only when using the symbolic formulation of the algorithm (presented in subsection 5.4.4).

The pessimistic and optimistic looping algorithms produce initial mean queue length estimates that respectively majorize and are majorized by the exact mean queue lengths. Since the MVA equations preserve majorization, the mean queue lengths produced after i applications of these equations starting from the initial estimates produced by the looping algorithms also respectively majorize and are majorized by the exact mean queue lengths. Theorem 5.6 utilizes this fact to establish bounding relationships between the performance measures produced by the LPBH algorithm, and the exact performance measures. (Theorem 3.3 cannot be applied directly here since the pessimistic initial mean queue length estimates do not necessarily bottleneck majorize the exact mean queue lengths.) In the theorem, the superscript (i) refers to the bound level (the looping algorithm iteration level is not explicitly indicated), and the subscripts UB and LB denote upper and lower bounds, respectively, as obtained from the LPBH algorithm as detailed in the theorem.

Theorem 5.6:

Consider an indexing of the fixed-rate service centers such that $L_k \geq L_{k+1}$, $1 \leq k \leq K-1$. For $N \geq i \geq 0$, $N \geq 1$,

[Primary Performance Measures]

$$R^{(i)}_{pess}(N) \geq R(N) \geq max[R^{(i)}_{opt}(N)\ ,\ a(N)] \tag{5-29}$$

$$\frac{N}{Z+R^{(i)}_{LB}(N)} \geq X(N) \geq X^{(i)}_{pess}(N) \tag{5-30}$$

$$X^{(i)}_{UB}(N)L_k \geq U_k(N) \geq U^{(i)}_{k\,pess}(N) \qquad 1 \leq k \leq K \tag{5-31}$$

[Secondary Performance Measures]

$$X_{JB}^{(i)}(N)Z \geq n_{K+1}(N) \geq n_{K+1\,pess}^{(i)}(N) \tag{5-32}$$

$$min\left[\frac{U_j^{(i)}{}_{JB}(N)}{1-U_j^{(i)}{}_{JB}(N)}\,,\; n_j^{(i)}{}_{opt}(N)\frac{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N)}{\sum_{k=1}^{j} n_k^{(i)}{}_{opt}(N)}\right]$$

$$\geq n_j(N) \geq max\left[n_j^{(i)}{}_{pess}(N)\frac{\sum_{k=1}^{j} n_k^{(i)}{}_{opt}(N)}{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N)}\,,\; U_k^{(i)}{}_{pess}(N)\right] \qquad 1\leq j\leq K \tag{5-33}$$

$$min\left[\frac{L_j}{1-U_j^{(i)}{}_{JB}(N)}\,,\; n_j^{(i)}{}_{opt}(N)\frac{\sum_{k=1}^{j} R_k^{(i)}{}_{pess}(N)}{\sum_{k=1}^{j} n_k^{(i)}{}_{opt}(N)}\right]$$

$$\geq R_j(N) \geq max\left[n_j^{(i)}{}_{pess}(N)\frac{\sum_{k=1}^{j} n_k^{(i)}{}_{opt}(N)}{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N)}\frac{1}{X_{JB}^{(i)}(N)}\,,\; L_j\right] \qquad 1\leq j\leq K \tag{5-34}$$

Proof:

The primary performance measure bounds, and the bounds in relation (5-32), follow directly from the correctness of the bound $a(N)$ and from the majorization relationships among the optimistic, pessimistic, and exact service center mean queue lengths. (The proof is the same as that used in theorem 3.3 for the lower bound on the mean system residence time, and the upper bounds on the system throughput, delay service center mean queue length, and fixed-rate service center utilizations.)

The derivation of the secondary performance measure bounds that use $U_k^{(i)}{}_{JB}(N)$ $(1\leq k\leq K)$ is identical to that of the corresponding bounds in theorem 3.3. The trivial lower bounds in relations (5-33) and (5-34) follow from $n_j(N) \geq U_j(N)$, and $R_j(N) \geq L_j$, respectively. The remaining bounds in relation (5-33) follow directly from (using the majorization relationships)

$$n_j^{(i)}{}_{opt}(N)\frac{\sum_{k=1}^{j} n_k(N)}{\sum_{k=1}^{j} n_k^{(i)}{}_{opt}(N)} \geq n_j(N) \geq n_j^{(i)}{}_{pess}(N)\frac{\sum_{k=1}^{j} n_k(N)}{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N)} \qquad 1\leq j\leq K$$

and (applying relation (3.2-3) in lemma 3.2)

$$\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N) \geq \sum_{k=1}^{j} n_k(N) \geq \sum_{k=1}^{j} n_k^{(i)}{}_{opt}(N) \qquad 1 \leq j \leq K$$

The remaining bounds in relation (5-34) are derived by applying Little's equation to relation (5-33). ∎

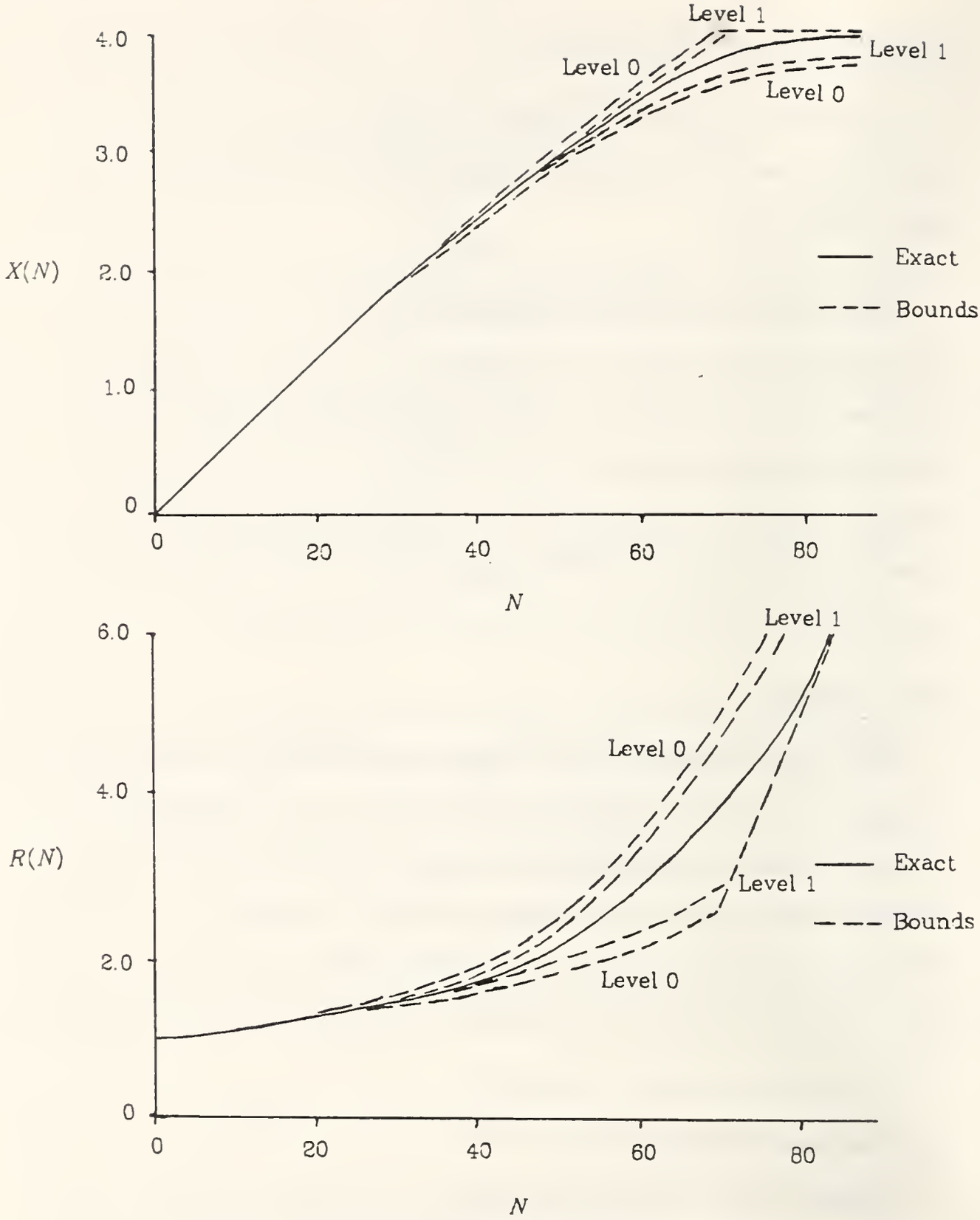
For queueing networks with open customer classes, in addition to the one closed class, LPBH bounds for the closed class are obtained by applying the algorithm to the closed queueing network with inflated closed class loadings. LPBH bounds for each open class s can be obtained from the closed class performance measure bounds as is done in the PBH algorithm (theorem 3.5), except that $L_{sc\ LB}^{h}$ must be taken as the minimum, over the fixed-rate service centers visited by the closed class, of the class s inflated loadings.

5.4.3. An Example

Table 5.5 gives the bound level 0, looping algorithm iteration level 5, LPBH bounds for the example queueing network of table 3.1, for a population of 50 customers. In figure 5.1, LPBH bounds on system throughput and mean system residence time are shown over a range of workload intensities. Level 0 and level 1 bounds are shown; a looping algorithm iteration level of 5 was used for both. Analogous results for the PBH algorithm are given in table 3.2 and figure 3.1.

5.4.4. Symbolic Formulation

To obtain primary performance measures, the symbolic formulation of the LPBH algorithm utilizes three major equations: one for the pessimistic looping algorithm, one for the optimistic looping algorithm, and one for the final pessimistic and optimistic computations. The latter equation is exactly that used in the symbolic formulation of the PBH algorithm:



LPBH Bounds on System Throughput and Mean System Residence Time for an Example Queueing Network

Figure 5.1

System Bounds					
System Throughput			Mean System Residence Time		
Upper	Exact	Lower	Upper	Exact	Lower
2.98	2.92	2.87	2.44	2.15	1.77

Fixed-Rate Service Center Bounds									
Service Center	Mean Residence Time			Mean Queue Length			Utilization		
	Upper	Exact	Lower	Upper	Exact	Lower	Upper	Exact	Lower
1	.98	.81	.39	2.93	2.35	1.18	.75	.73	.72
2	.50	.46	.23	1.48	1.33	.69	.60	.58	.57
3	.34	.33	.20	1.03	.95	.59	.51	.50	.49
4	.31	.29	.19	.91	.85	.58	.48	.47	.46
5	.11	.10	.08	.31	.30	.23	.24	.23	.23
6	.07	.07	.06	.22	.21	.17	.18	.17	.17
7	.06	.06	.05	.18	.17	.14	.15	.15	.14
8	.03	.03	.03	.10	.10	.09	.09	.09	.09

Delay Service Center Bounds		
Mean Queue Length		
Upper	Exact	Lower
44.73	43.74	43.00

Bound Level 0, Iteration Level 5 LPBH Bounds for an Example Queueing Network

Table 5.5

$$W^{(j)}(n,p) = L^{(p)} + \frac{W^{(j-1)}(n,p+1)}{Z + W^{(j-1)}(n-p,1)}(n-p) \qquad j \geq 1 \qquad (5\text{-}35)$$

where

$$W^{(j)}_{pess}(n,p) = \sum_{k=1}^{K} L_k^{p-1} R^{(j)}_{k\,pess}(n-p+1)$$

and

$$W^{(j)}_{opt}(n,p) = \sum_{k=1}^{K} L_k^{p-1} R^{(j)}_{k\,opt}(n-p+1)$$

Equation (5-35) is used to compute $W^{(i)}_{pess}(N,1)$ and $W^{(i)}_{opt}(N,1)$, starting from $W^{(0)}_{pess}(n,p)$ and $W^{(0)}_{opt}(n,p)$ quantities, respectively, such that $(N-1)-i=n-p$

$(1 \le p \le i+1)$. These latter quantities are given by

$$W^{(0)}_{pess}(n,p) = \sum_{k=1}^{K} L_k^{p-1} R^{(0)}_{k\,pess}(N-i) = V^{(l_{pess})}_{pess}(N-i,p)$$

and

$$W^{(0)}_{opt}(n,p) = \sum_{k=1}^{K} L_k^{p-1} R^{(0)}_{k\,opt}(N-i) = V^{(l_{opt})}_{opt}(N-i,p)$$

where l_{pess} and l_{opt} denote the number of iterations used in the pessimistic and optimistic looping algorithms, respectively.

$$V^{(m)}_{pess}(N-i,q) = \sum_{k=1}^{K} L_k^{q-1} R^{(m)}_{k\,pess}(N-i)$$

(where the superscript (m) now denotes quantities yielded by m looping algorithm iterations) and

$$V^{(m)}_{opt}(N-i,q) = \sum_{k=1}^{K} L_k^{q-1} R^{(m)}_{k\,opt}(N-i)$$

$V^{(l_{pess})}_{pess}(N-i,p)$ $(1 \le p \le i+1)$ is computed iteratively using, for $m \ge 1$,

$$V^{(m)}_{pess}(N-i,q) = L^{(q)} + \frac{V^{(m-1)}_{pess}(N-i,q+1)}{Z + V^{(m-1)}_{pess}(N-i,1)}(N-i-1) \qquad (5\text{-}36)$$

starting from $V^{(0)}_{pess}(N-i,q)$ quantities such that $1 \le q \le i+1+l_{pess}$. Equation (5-36) is derived from the equations defining the pessimistic looping algorithm. From the choice of initial performance measure estimates,

$$V^{(0)}_{pess}(N-i,q) = L_b^{q-1} b(N-i)$$

$V^{(l_{opt})}_{opt}(N-i,p)$ $(1 \le p \le i+1)$ is computed iteratively using, for $m \ge 1$,

$$V^{(m)}_{opt}(N-i,q) = L^{(q)} + \frac{V^{(m-1)}_{opt}(N-i,q+1) - \frac{L^{(N-i+q)}}{L^{(N-i-1)}}\Delta^{(m-1)}_{B}(N-i)}{Z + V^{(m-1)}_{opt}(N-i,1) - \frac{L^{(N-i)}}{L^{(N-i-1)}}\Delta^{(m-1)}_{B}(N-i)}(N-i-1) \qquad (5\text{-}37)$$

where

$$\Delta^{(m)}_{B}(N-i) = \frac{V^{(m)}_{opt}(N-i,1)}{Z + V^{(m)}_{opt}(N-i,1)}(N-i) - \frac{V^{(m)}_{opt}(N-i,1) - L_b}{Z + V^{(m)}_{opt}(N-i,1) - L_b}(N-i-1)$$

starting from $V^{(0)}_{opt}(N-i,q)$ quantities such that $1 \le q \le i+1+l_{opt}$. Equation (5-37) is derived from the equations defining the optimistic looping algorithm. From

the choice of initial performance measure estimates,

$$V_{opt}^{(0)}(N-i,q) = L^{(q)}(1+\frac{\alpha(N-i-1)}{Z+\alpha(N-i-1)}(N-i-1))$$

For bound level i, l_{pess} pessimistic looping algorithm iterations, and l_{opt} optimistic looping algorithm iterations, $L^{(p)}$ must be calculated for $1 \le p \le i+max[l_{pess}, l_{opt}+1]$, and $L^{(N-i+p)}$ must be calculated for $-1 \le p \le i+1+l_{opt}$. For queueing networks with many fixed-rate service centers, this is the dominant computational requirement of the symbolic formulation of the LPBH algorithm. The major advantage of the symbolic formulation is that these computations can be performed in a separate initial step. This allows the bound level to be chosen dynamically, although there is a relatively small computational penalty in so doing, and reduces the expense of calculating bounds for multiple population levels. The computational costs of the two formulations will be considered in detail in subsection 5.4.5.

To obtain secondary performance measures for each fixed-rate service center k of interest, three major additional equations are required: one for the pessimistic looping algorithm, one for the optimistic looping algorithm, and one for the final pessimistic and optimistic computations. The latter equation is exactly that used in the symbolic formulation of the PBH algorithm:

$$W_k^{(j)}(n,p) = L_k^p + \frac{W_k^{(j-1)}(n,p+1)}{Z+W^{(j-1)}(n-p,1)}(n-p) \qquad j \ge 1 \tag{5-38}$$

where $W_{k\,pess}^{(j)}(n,p) = L_k^{p-1} R_{k\,pess}^{(j)}(n-p+1)$ and $W_{k\,opt}^{(j)}(n,p) = L_k^{p-1} R_{k\,opt}^{(j)}(n-p+1)$. Equation (5-38) is used to compute $W_{k\,pess}^{(i)}(N,1)$ and $W_{k\,opt}^{(i)}(N,1)$, starting from $W_{k\,pess}^{(0)}(N,i+1)$ and $W_{k\,opt}^{(0)}(N,i+1)$, respectively. These latter quantities are given by

$$W_{k\,pess}^{(0)}(N,i+1) = L_k^i R_{k\,pess}^{(0)}(N-i) = V_{k\,pess}^{(l_{pess})}(N-i,i+1)$$

and

$$W_{k\,opt}^{(0)}(N,i+1) = L_k^i R_{k\,opt}^{(0)}(N-i) = V_{k\,opt}^{(l_{opt})}(N-i,i+1)$$

where $V_{k\ pess}^{(m)}(N-i,q) = L_k^{q-1} R_{k\ pess}^{(m)}(N-i)$ (again, the superscript (m) now denotes quantities yielded by m looping algorithm iterations) and $V_{k\ opt}^{(m)}(N-i,q) = L_k^{q-1} R_{k\ opt}^{(m)}(N-i)$.

$V_{k\ pess}^{(l_{pess})}(N-i,i+1)$ is computed iteratively using, for $m \geq 1$,

$$V_{k\ pess}^{(m)}(N-i,q) = L_k^q + \frac{V_{k\ pess}^{(m-1)}(N-i,q+1)}{Z + V_{pess}^{(m-1)}(N-i,1)}(N-i-1) \qquad (5\text{-}39)$$

starting from $V_{k\ pess}^{(0)}(N-i,i+1+l_{pess})$. From the choice of initial performance measure estimates,

$$V_{k\ pess}^{(0)}(N-i,i+1+l_{pess}) = \begin{cases} L_b^{i+l_{pess}} b\,(N-i) & \text{for } k=b \\ 0 & \text{for } k \neq b,\ 1 \leq k \leq K \end{cases}$$

$V_{k\ opt}^{(l_{opt})}(N-i,i+1)$ is computed iteratively using, for $m \geq 1$,

$$V_{k\ opt}^{(m)}(N-i,q) = L^{(q)} + \frac{V_{k\ opt}^{(m-1)}(N-i,q+1) - \frac{L_k^{N-i+q}}{L^{(N-i-1)}}\Delta_B^{(m-1)}(N-i)}{Z + V_{opt}^{(m-1)}(N-i,1) - \frac{L^{(N-i)}}{L^{(N-i-1)}}\Delta_B^{(m-1)}(N-i)}(N-i-1) \qquad (5\text{-}40)$$

starting from $V_{k\ opt}^{(0)}(N-i,i+1+l_{opt})$. From the choice of initial performance measure estimates,

$$V_{k\ opt}^{(0)}(N-i,i+1+l_{opt}) = L_k^{i+1+l_{opt}}\left(1 + \frac{a(N-i-1)}{Z+a(N-i-1)}(N-i-1)\right)$$

5.4.5. Computational Cost

Table 5.6 summarizes results concerning the computational complexity of both the symbolic and the non-symbolic formulations of the LPBH algorithm. The quantities given are the dominant terms in the space and time complexities of computing bound level i LPBH performance measures (either pessimistic, optimistic, or both) with, where applicable, l_{pess} pessimistic looping algorithm iterations, and l_{opt} optimistic looping algorithm iterations. In the table, l_{max} denotes $max[l_{pess}\ ,\ l_{opt}+1]$.

The derivation of these complexities is similar to the derivation of the

Non-symbolic Formulation			
	Pessimistic	Optimistic	Both
Time	$4K(i+l_{pess})$	$4Ki+6Kl_{opt}$	$8Ki+4Kl_{pess}+6Kl_{opt}$
Space	K	$2K$	$2K$

Symbolic Formulation			
	Pessimistic	Optimistic	Both
Time	$(i+l_{pess})^2+$ $[2K(i+l_{pess})]+$ $2k(i+l_{pess})$	$(i+l_{opt})^2+l_{opt}(2i+l_{opt})+$ $[4K(i+l_{opt})]+$ $2k(i+l_{opt})+2kl_{opt}$	$(i+l_{pess})^2+(i+l_{opt})^2+l_{opt}(2i+l_{opt})+$ $[2K(i+l_{max})+2K(i+l_{opt})]+$ $2k(i+l_{pess})+2k(i+l_{opt})+2kl_{opt}$
Space	$2(i+l_{pess})+$ $k(i+l_{pess})$	$3(i+l_{opt})+k(i+l_{opt})+$ kl_{opt}	$2(i+l_{max})+(i+l_{opt})+$ $k(i+l_{max})+kl_{opt}$

k = number of fixed-rate service centers for which secondary performance measures are required

The Computational Cost of the LPBH Algorithm

Table 5.6

corresponding quantities for the single closed class PBH algorithm. In fact, the complexity of computing bound level i pessimistic LPBH performance measures with l pessimistic looping algorithm iterations is the same as that of computing bound level $i+l$ pessimistic PBH performance measures. Optimistic LPBH performance measures are more costly due to the increased complexity of the optimistic looping algorithm. The per-iteration time complexities of the non-symbolic and symbolic formulations of the optimistic looping algorithm are higher by $2K$ and 2, respectively. In addition, the non-symbolic and symbolic formulations require additional storage for the L_k^{N-i} $(1\le k\le K)$ and $L^{(N-i+q)}$ $(-1\le q\le i+1+l_{opt})$ values, respectively. (The space complexity of the symbolic formulation is also increased by the need to store L_k^{N-i+q} values, for

$i+1 \leq q \leq i+1+l_{opt}$, when obtaining secondary performance measures for service center k.) Finally, the computation of the $L^{(N-i+q)}$ values makes a significant contribution to the time complexity.

If bounds at multiple bound hierarchy levels are computed, the symbolic formulation of the LPBH algorithm does not have to repeat the computations represented by the terms in square brackets in table 5.6, while the non-symbolic formulation must repeat everything. There are also computational savings with the symbolic formulation if bounds for the same queueing network with different population levels are computed, as the computations represented by the $2K(i+l_{pess})$ and $2K(i+l_{max})$ terms, and $2K(i+l_{opt})$ of the $4K(i+l_{opt})$ term, do not have to be repeated. The primary disadvantage of the symbolic formulation is the cost of obtaining secondary performance measures for more than a few of the queueing network service centers.

5.4.6. Algorithm Accuracy

The accuracy of the single closed class LPBH algorithm has been studied by applying the algorithm to the queueing networks used in testing the single closed class PBH algorithm. Each queueing network was analyzed with four population sizes; these were chosen to be the smallest populations that yielded exact bottleneck service center utilizations of at least 50%, 70%, 85% and 95%. For looping algorithm iteration levels of 3, 5, 7 and 50, level 0 LPBH bounds were obtained for each queueing network for each of its population levels. (Recall that the iteration level only sets a limit on the number of iterations to be performed; fewer iterations are performed if convergence occurs before reaching the limit.) Over each of several chosen groupings of tests, the maximum and the average of the maximal errors for system throughput and mean system residence time were computed for each looping algorithm iteration level.

Table 5.7 summarizes the results for the 192 combinations of queueing network and bottleneck service center utilization. Note that the maximum and the average of the maximal errors decrease with increasing looping algorithm iteration level. Also, they are higher for the mean system residence time than for the system throughput.

In tables 5.8 to 5.11, only the level 0 bounds obtained with a looping algorithm iteration level of 5 are considered. In each table, a particular parameter has been chosen (respectively, the exact bottleneck utilization, the delay service center loading, the fixed-rate service center loading distribution skewness, and the number of fixed-rate service centers), and the behavior of the maximal errors to parameter changes has been illustrated. This behavior is usually quite similar to that exhibited by the PBH algorithm (tables 3.7 to 3.11). The maximum and the average of the maximal errors increase as the skewness increases and as the exact bottleneck utilization increases. (This last effect is dependent on the bound and iteration level and on the range of utilizations considered. As in the PBH algorithm, the maximal errors are asymptotically zero as the queueing network population increases.) The maximum and the average of the maximal errors in the mean system residence time increase as the delay service center loading increases, while increasing and then decreasing for the system throughput. Finally, the average maximal errors decrease as the number of fixed-rate service centers increase, while the maximum maximal errors increase. This behavior is somewhat different than that exhibited by the PBH algorithm, and is likely due to the properties of the optimistic looping algorithm.

Looping Algorithm Iteration Level	Maximal Errors (%)			
	$R(N)$ Aver.	$R(N)$ Max.	$X(N)$ Aver.	$X(N)$ Max.
3	11.4	38.8	3.8	8.3
5	7.3	31.2	2.4	5.7
7	5.1	27.7	1.7	4.8
50	1.7	15.5	0.6	2.9

Maximal Errors for a Range of Looping Algorithm Iteration Levels

Table 5.7

Bottleneck Utilization	Maximal Errors (Looping Algorithm Iteration Level of 5) (%)			
	$R(N)$ Aver.	$R(N)$ Max.	$X(N)$ Aver.	$X(N)$ Max.
.50	1.4	4.1	0.4	1.4
.70	5.0	13.7	1.3	3.2
.85	10.5	28.6	3.1	4.5
.95	12.2	31.2	4.7	5.7

Maximal Errors for a Range of Bottleneck Utilizations

Table 5.8

Delay Service Center Loading	Maximal Errors (Looping Algorithm Iteration Level of 5) (%)			
	$R(N)$ Aver.	$R(N)$ Max.	$X(N)$ Aver.	$X(N)$ Max.
0	2.3	5.1	2.3	5.1
10	10.1	22.2	2.6	5.7
20	14.4	31.2	2.3	5.5

Maximal Errors for a Range of Delay Service Center Loadings

Table 5.9

Loading Skewness (value of p)	Maximal Errors (Looping Algorithm Iteration Level of 5) (%)			
	$R(N)$ Aver.	$R(N)$ Max.	$X(N)$ Aver.	$X(N)$ Max.
0.5	6.1	24.0	2.1	5.4
1.0	7.4	29.3	2.4	5.6
2.0	8.3	31.2	2.6	5.7

Maximal Errors for a Range of p Values

Table 5.10

Number of Fixed-Rate Service Centers	Maximal Errors (Looping Algorithm Iteration Level of 5) (%)			
	$R(N)$ Aver.	$R(N)$ Max.	$X(N)$ Aver.	$X(N)$ Max.
20	7.6	30.5	2.4	5.6
60	7.0	31.2	2.3	5.7

Maximal Errors for two Different Queueing Network Sizes

Table 5.11

5.5. A Looped Multiple Closed Class Performance Bound Hierarchy Algorithm

5.5.1. The Looping Algorithm

The looping algorithm that is utilized by the multiple closed class LPBH algorithm is based on the following equation giving the population level $\vec{N}-\vec{e}_s$ class r mean queue length at a fixed-rate service center k in terms of that for population level $\vec{N}$:

$$n_{kr}(\vec{N}-\vec{e}_s) = \frac{X_s^{+k}(\vec{N}-\vec{e}_r)}{X_s(\vec{N})} n_{kr}(\vec{N}) \tag{5-41}$$

This equation is derived from the Convolution solution algorithm expressions for system throughput and mean queue length [Zahorjan 80]. To be useful in a looping algorithm, the class s system throughput in the queueing network resulting from the addition of a service center with loadings identical to those of service center k ($X_s^{+k}(\vec{N}-\vec{e}_r)$) must be replaced by an approximation to this quantity. Utilizing Little's equation, and the level 0 multiple closed class PBH algorithm upper bound on mean system residence time (distinguished by the subscript UB and the superscript $PBH(0)$), yields the following approximation:

$$\tilde{n}_{kr}(\vec{N}-\vec{e}_s) = \frac{N_s-\delta_{rs}}{N_s} \frac{Z_s+R_s(\vec{N})}{Z_s+R_s^{+k}{}_{UB}^{PBH(0)}(\vec{N}-\vec{e}_r)} n_{kr}(\vec{N})$$

The multiple closed class PBH algorithm equations are used to complete the looping, yielding, for $s\in C$ and $i\geq 1$,

$$R_{ks}^{(i)}(\vec{N}) = L_{ks}\left(1+\sum_{r\in C}\tilde{n}_{kr}^{(i-1)}(\vec{N}-\vec{e}_s)\right) \qquad 1\leq k\leq K \tag{5-42}$$

$$R_{s\,opt}^{(i)}(\vec{N}) = max\left[a_s^{(i)}(\vec{N}),\ \sum_{k=1}^{K} R_{ks}^{(i)}(\vec{N}) + \sum_{r\in C} L_{sr}^{h}{}_{LB}\, n_{r\,opt}^{h(i-1)}(\vec{N}-\vec{e}_s)\right] \tag{5-43}$$

$$R_{s\,pess}^{(i)}(\vec{N}) = \sum_{k=1}^{K} R_{ks}^{(i)}(\vec{N}) + \sum_{r\in C} L_{sr}^{h}{}_{UB}\, n_{r\,pess}^{h(i-1)}(\vec{N}-\vec{e}_s) \tag{5-44}$$

$$n_{ks}^{(i)}(\vec{N}) = \frac{R_{ks}^{(i)}(\vec{N})}{Z_s+R_{s\,pess}^{(i)}(\vec{N})} N_s \qquad 1\leq k\leq K \tag{5-45}$$

$$\tilde{n}_{kr}^{(i)}(\vec{N}-\vec{e}_s) = \frac{N_s-\delta_{rs}}{N_s} \frac{Z_s+R_{s\,opt}^{(i)}(\vec{N})}{Z_s+R_s^{+k}{}_{UB}^{PBH(0)}(\vec{N}-\vec{e}_r)} n_{kr}^{(i)}(\vec{N}) \tag{5-46}$$

$$n_{r\,opt}^{h(i)}(\vec{N}-\vec{e}_s) = max\left[0,\ \frac{R_{r\,LB}^{PBH(0)}(\vec{N}-\vec{e}_s)}{Z_r+R_{r\,LB}^{PBH(0)}(\vec{N}-\vec{e}_s)}(N_r-\delta_{rs}) - \sum_{k=1}^{K}\tilde{n}_{kr}^{(i)}(\vec{N}-\vec{e}_s)\right] \tag{5-47}$$

and

$$n_{r\,pess}^{h\,(i)}(\vec{N}-\vec{e}_s) = \frac{R_r^{PBH(0)}{}_{UB}(\vec{N}-\vec{e}_s)}{Z_r + R_r^{PBH(0)}{}_{UB}(\vec{N}-\vec{e}_s)}(N_r - \delta_{rs}) - \sum_{k=1}^{K} \tilde{n}_{kr}^{(i)}(\vec{N}-\vec{e}_s) \qquad (5\text{-}48)$$

where the superscript (i) denotes level i multiple class looping algorithm quantities produced from initial performance measure estimates superscripted by (0), $a_s^{(i)}(\vec{N})$ is defined as in the multiple closed class PBH algorithm (equation (4-7)), and $L_{sr\,UB}^{h}$ and $L_{sr\,LB}^{h}$ are chosen to be the maximum and the minimum, respectively, over the fixed-rate service centers visited by class r, of the class s loadings. Equations (5-46) to (5-48) utilize level 0 multiple closed class PBH algorithm bounds; these bounds could be replaced with others if desired. In the numerical tests and examples of the LPBH algorithm that will be presented here, a tighter bound was used in place of $R_r^{PBH(0)}{}_{LB}(\vec{N}-\vec{e}_s)$ in equation (5-47). This bound was derived by calculating the level 1 PBH lower bound with $L_{sr\,LB}^{h}$ chosen as in the looping algorithm, and with $a_s^{(0)}(\vec{N}-\vec{e}_s)$ used in place of $a_s^{(1)}(\vec{N}-\vec{e}_s)$. One suitable choice of initial estimates for each class s uses those for the PBH algorithm:

$$R_{ks}^{(0)}(\vec{N}) = n_{ks}^{(0)}(\vec{N}) = 0 \qquad (5\text{-}49)$$

$$R_{s\,pess}^{(0)}(\vec{N}) = b_s(\vec{N}) \qquad (5\text{-}50)$$

and

$$R_{s\,opt}^{(0)}(\vec{N}) = a_s^{(0)}(\vec{N}) \qquad (5\text{-}51)$$

For each class r, $n_{r\,opt}^{h\,(0)}(\vec{N}-\vec{e}_s)$, $n_{r\,pess}^{h\,(0)}(\vec{N}-\vec{e}_s)$, and $\tilde{n}_{kr}^{(0)}(\vec{N}-\vec{e}_s)$ $(1 \le k \le K)$, are computed using equations (5-46) to (5-48).

It is proved in theorem 5.7 that the chosen initial performance measure estimates guarantee convergence to a solution of the multiple class looping algorithm equations. This convergence is monotonic in the sense that each of the performance measure values produced at the $i+1$st iteration lies in the interval defined by that produced at the ith iteration and that of the exact solution to the equations.

Theorem 5.7:

The initial performance measure estimates given by equations (5-49) to (5-51) guarantee monotonic convergence to a solution of the multiple class looping algorithm equations.

Proof:

It is sufficient to establish that, for all $i \geq 0$, $s \in C$,

$$n_{ks}^{(i)}(\vec{N}) \leq n_{ks}^{(i+1)}(\vec{N}) \qquad 1 \leq k \leq K$$

$$n_{s\,pess}^{h\,(i)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{(i)}(\vec{N}) \geq n_{s\,pess}^{h\,(i+1)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{(i+1)}(\vec{N})$$

and

$$n_{s\,opt}^{h\,(i)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{(i)}(\vec{N}) \leq n_{s\,opt}^{h\,(i+1)}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{(i+1)}(\vec{N})$$

where $n_{s\,pess}^{h\,(i)}(\vec{N})$ and $n_{s\,opt}^{h\,(i)}(\vec{N})$ are defined as in the PBH algorithm (equations (4-5) and (4-6)).

These relations are established for $i=0$ by noting that $n_{ks}^{(0)}(\vec{N}) = 0 \leq n_{ks}^{(1)}(\vec{N})$, $R_{s\,pess}^{(0)}(\vec{N}) \geq R_{s\,pess}^{(1)}(\vec{N})$, and $R_{s\,opt}^{(0)}(\vec{N}) \leq R_{s\,opt}^{(1)}(\vec{N})$. The theorem then follows from lemma 5.10 in appendix 5.A, which implies that the optimistic and pessimistic properties of a mean queue length allocation and its associated heaps are preserved by the multiple class looping algorithm equations. ∎

It is shown in theorem 5.8 that the performance measures yielded by the multiple class looping algorithm are suitable as initial performance measure estimates for the multiple closed class PBH algorithm. In particular, it is proved that the resulting mean queue length allocation, in comparison to the exact mean queue length allocation, is heap-optimistic and heap-pessimistic in conjunction with its optimistic and pessimistic heaps, respectively (where the exact allocation is treated as having heaps equal to zero).

Theorem 5.8:

For $i \geq 0$, $s \in C$,

$$n_{ks}^{(i)}(\vec{N}) \leq n_{ks}(\vec{N}) \tag{5-52}$$

$$n_s^{h(i)}{}_{pess}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{(i)}(\vec{N}) \geq \sum_{k=1}^{K} n_{ks}(\vec{N}) \tag{5-53}$$

and

$$n_s^{h(i)}{}_{opt}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{(i)}(\vec{N}) \leq \sum_{k=1}^{K} n_{ks}(\vec{N}) \tag{5-54}$$

Proof:

The proof is by induction on i.

For $i=0$, the theorem holds from the choice of the initial performance measure estimates. Assume that the theorem holds for i, and consider the theorem for $i+1$ ($i \geq 0$). Relation (5-53) will be established if it can be shown that, for each $s \in C$,

$$R_s^{(i+1)}{}_{pess}(\vec{N}) \geq R_s(\vec{N}) \tag{5-55}$$

which reduces to, using equation (5-44) and the definition of $R_s(\vec{N})$,

$$\sum_{k=1}^{K} R_{ks}^{(i+1)}(\vec{N}) + \sum_{r \in C} L_{sr}^{h}\, {}_{JB}n_r^{h(i)}{}_{pess}(\vec{N}-\vec{e}_s) \geq \sum_{k=1}^{K} R_{ks}(\vec{N})$$

From equation (5-42) and the MVA mean residence time equation, this relation can be expanded to yield, after regrouping terms and canceling $\sum_{k=1}^{K} L_{ks}$ from both sides of the relation,

$$\sum_{r \in C} L_{sr}^{h}\, {}_{JB}\, n_r^{h(i)}{}_{pess}(\vec{N}-\vec{e}_s) \geq \sum_{r \in C} \sum_{k=1}^{K} L_{ks}\left(n_{kr}(\vec{N}-\vec{e}_s) - \tilde{n}_{kr}^{(i)}(\vec{N}-\vec{e}_s)\right) \tag{5-56}$$

From the inductive hypothesis on relation (5-54), $R_s^{(i)}{}_{opt}(\vec{N}) \leq R_s(\vec{N})$. In conjunction with the inductive hypothesis on relation (5-52), and the correctness of the PBH algorithm, this yields

$$\frac{N_s - \delta_{rs}}{N_s}\,\frac{Z_s + R_s(\vec{N})}{Z_s + R_s^{+k}(\vec{N}-\vec{e}_r)}\, n_{kr}(\vec{N}) \geq \frac{N_s - \delta_{rs}}{N_s}\,\frac{Z_s + R_s^{(i)}{}_{opt}(\vec{N})}{Z_s + R_s^{+k}{}_{JB}^{PBH(0)}(\vec{N}-\vec{e}_r)}\, n_{kr}^{(i)}(\vec{N})$$

Using equation (5-41) in conjunction with Little's equation, and equation (5-46), this last relation yields

$$n_{kr}(\vec{N}-\vec{e}_s) \geq \tilde{n}_{kr}^{(i)}(\vec{N}-\vec{e}_s) \tag{5-57}$$

implying that each term in the double summation on the right-hand side of relation (5-56) is non-negative. These terms represent an allocation of a heap for each class r, given by $\sum_{k=1}^{K} n_{kr}(\vec{N}-\vec{e}_s) - \sum_{k=1}^{K} \tilde{n}_{kr}^{(i)}(\vec{N}-\vec{e}_s)$, to the fixed-rate service centers. Since, by construction, $L_{sr}^{h}{}_{UB}$ is an upper bound on the mean class s loading when encountering customers from the class r heap, the relation holds if it holds with L_{ks} replaced by $L_{sr}^{h}{}_{UB}$. Making this substitution, separating the terms corresponding to each class, and simplifying, results in a relation that holds from the correctness of the PBH algorithm bound used in equation (5-48). This establishes relation (5-53) for $i+1$.

From relation (5-57), the MVA equations, and equations (5-42) and (5-45), relation (5-55) also establishes relation (5-52) for $i+1$. Relation (5-54) will be established for $i+1$ if it can be shown that, for each $s \in C$,

$$R_s^{(i+1)}{}_{opt}(\vec{N}) \leq R_s(\vec{N}) \tag{5-58}$$

From relation (5-55) and the definition of $a_s(\vec{N})$, $a_s^{(i+1)}(\vec{N}) \leq R_s(\vec{N})$. Therefore, from equation (5-43) and the definition of $R_s(\vec{N})$, relation (5-58) will be shown if it can be shown that

$$\sum_{k=1}^{K} R_{ks}^{(i+1)}(\vec{N}) + \sum_{r \in C} L_{sr}^{h}{}_{LB} n_r^{h}{}_{opt}^{(i)}(\vec{N}-\vec{e}_s) \leq \sum_{k=1}^{K} R_{ks}(\vec{N})$$

From equation (5-42) and the MVA mean residence time equation, this relation can be expanded to yield, after regrouping terms and canceling $\sum_{k=1}^{K} L_{ks}$ from both sides of the relation,

$$\sum_{r \in C} L_{sr}^{h}{}_{LB}\, n_r^{h}{}_{opt}^{(i)}(\vec{N}-\vec{e}_s) \leq \sum_{r \in C} \sum_{k=1}^{K} L_{ks}\left(n_{kr}(\vec{N}-\vec{e}_s) - \tilde{n}_{kr}^{(i)}(\vec{N}-\vec{e}_s)\right)$$

From relation (5-57), each term in the double summation on the right-hand

side of this last relation is non-negative. These terms represent an allocation of a heap for each class r, given by $\sum_{k=1}^{K} n_{kr}(\vec{N}-\vec{e}_s) - \sum_{k=1}^{K} \tilde{n}_{kr}^{(i)}(\vec{N}-\vec{e}_s)$, to the fixed-rate service centers. Since, by construction, $L_{sr\ LB}^{h}$ is a lower bound on the mean class s loading when encountering customers from the class r heap, the relation holds if it holds with L_{ks} replaced by $L_{sr\ LB}^{h}$. Making this substitution, separating the terms corresponding to each class, and simplifying, results in a relation that holds from the correctness of the PBH algorithm bound used in equation (5-47). This establishes the theorem. ∎

5.5.2. The LPBH Algorithm Definition

The multiple closed class LPBH algorithm has two computational phases. In the first phase, initial performance measure estimates for each population resulting from the removal of i customers, where i is the desired bound level, are computed using the multiple class looping algorithm of the previous subsection. In the second phase, performance measures for the original population level are obtained by applying the multiple closed class PBH algorithm equations to the initial estimates obtained from phase one. These equations are applied with $L_{sr\ UB}^{h}$ chosen as previously, and $L_{sr\ LB}^{h}$ chosen as the minimum, over the fixed-rate service centers visited by class r, of the class s loadings.

The first phase provides a tradeoff between computational effort and accuracy since any number of iterations may be used in each application of the looping algorithm. As with the single closed class LPBH algorithm, it will be assumed that there is an iteration level that limits the number of iterations to be performed in each application.

The iteration level can be chosen dynamically, depending on the tightness of the initial performance measure estimates being produced. It is not computationally efficient to choose the bound level dynamically. (There is no

computationally efficient symbolic formulation of the algorithm that has been developed.)

From theorem 5.8, and since the optimistic and pessimistic properties of a mean queue length allocation and its associated heaps are preserved by the multiple closed class PBH algorithm equations, the performance measures yielded by the second computational phase satisfy relations (4-11) to (4-13) of theorem 4.1 (replacing the level $i+1$ performance measures by the exact performance measures). (Note that an application of the exact MVA equations is identical in effect to an application of the PBH algorithm equations with the same mean queue length values, and with zero heaps.) Theorem 4.2 can therefore be applied, with A_k replaced by 1 for $1 \leq k \leq K$, to obtain bounds on primary and secondary performance measures. In queueing networks with open customer classes, theorem 4.3 can be applied to obtain bounds on open class performance measures.

The multiple closed class PBH algorithm can be extended to treat queueing networks with load-dependent service centers, as was detailed in the previous chapter. The multiple closed class LPBH algorithm can be similarly extended by using the PBH algorithm equations required for load dependence in the second computational phase, and by treating each load-dependent service center k as a fixed-rate service center with class s loading $\frac{L_{ks}(c_k)}{c_k}$ in the multiple class looping algorithm service center mean residence time and queue length computations of the first phase. Also, in both phases the computation of $R_{s\,pess}^{(i)}(\vec{N})$ and $a_s^{(i)}(\vec{N})$, and the choices of $L_{ST\ UB}^{h}$ and $L_{ST\ LB}^{h}$, must be modified to reflect the presence of load-dependent service centers; this is done in the same manner as in the PBH algorithm. With these modifications, the multiple class looping algorithm yields all of the performance measures estimates required for the second phase, except for the $p_k(m \mid \vec{N})$ values, for $1 \leq m \leq c_k - 2$, for a

load-dependent service center k. These latter values must be taken as zero, as in the PBH algorithm.

5.5.3. An Example

Table 5.12 gives level 3 LPBH bounds for the queueing network of table 4.1, for a class T arrival rate of .1, 3 class B customers, and 35 class I customers. Empirically, the multiple class looping algorithm converges very quickly; in this example a looping algorithm iteration level of 10 was used, and convergence had already occurred after about 7 iterations. As in table 4.2, results are displayed using the time unit choice for each class that yields fixed-rate service center loadings of the class that sum to unity. Using the same time unit choices, figure 5.2 gives LPBH bounds on the class I system throughput and mean system residence time over a range of class I workload intensities, again using a looping algorithm iteration level of 10. Analogous results for the PBH algorithm are given in table 4.2 and figure 4.1.

5.5.4. Computational Cost

Table 5.13 summarizes results concerning the computational complexity of the multiple closed class LPBH algorithm. The quantities given are the dominant terms in the space and time complexities of computing level i bounds, when an average of l looping algorithm iterations are performed in the applications of the multiple class looping algorithm, for population level $\vec{N}$, assuming that $N_r \geq i$ for all $r \in C$. These results follow directly from the space complexity ($K|C|$) and time complexity ($6K|C|$) of an application of the multiple closed class PBH algorithm equations, and the time complexity ($6K|C|+K|C|^2$) of an iteration using the multiple class looping algorithm equations.

System Throughput Bounds					
Class B			Class I		
Upper	Exact	Lower	Upper	Exact	Lower
1.47	1.44	1.39	1.07	1.06	1.06

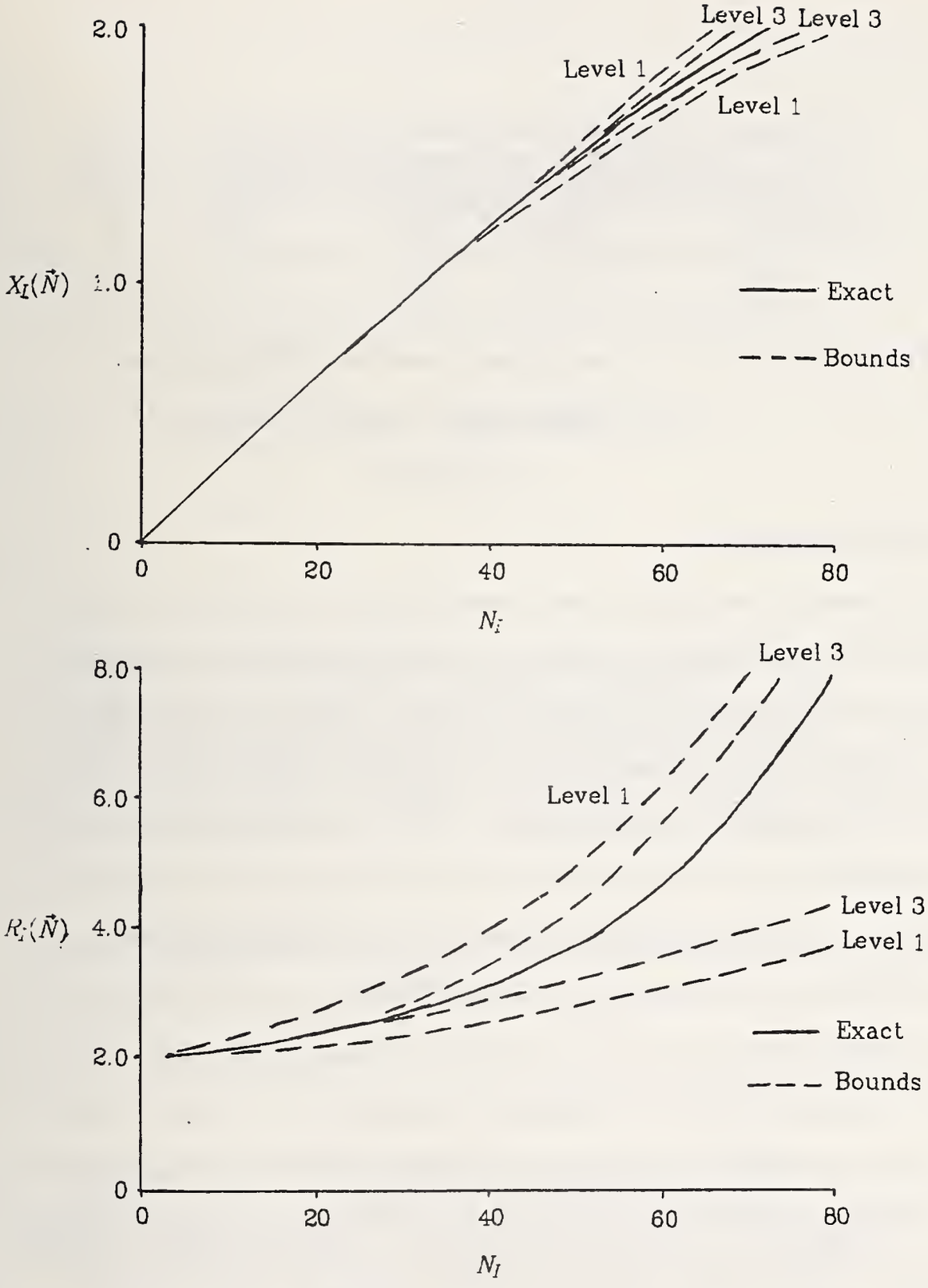
Mean System Residence Time Bounds								
Class T			Class B			Class I		
Upper	Exact	Lower	Upper	Exact	Lower	Upper	Exact	Lower
3.27	3.09	2.98	2.16	2.08	2.04	3.11	2.90	2.74

Fixed-Rate Service Center Utilization Bounds									
Service Center	Class B			Class I			Total over all classes		
	Upper	Exact	Lower	Upper	Exact	Lower	Upper	Exact	Lower
1	.14	.13	.13	.21	.21	.21	.85	.85	.84
2	.07	.07	.06	.11	.11	.11	.58	.57	.57
3	.18	.17	.17	.05	.05	.05	.57	.57	.56
4	04	.04	04	.27	.27	26	.63	.63	62
5	.23	.23	.22	.06	.06	.06	.46	.45	.44
6	.26	.25	.24	.18	.18	.18	.56	.55	.54
7	.34	.33	.32	.15	.15	.15	.59	.58	.57
8	.22	.21	.21	.03	.03	.03	.31	.31	.30

Fixed-Rate Service Center Mean Queue Length Bounds									
Service Center	Class T			Class B			Class I		
	Upper	Exact	Lower	Upper	Exact	Lower	Upper	Exact	Lower
1	3.22	2.82	2.63	.77	.65	.59	1.45	1.17	1.04
2	1.31	.93	.92	.32	.15	.14	.65	.25	.24
3	1.05	.76	.75	.52	.35	.34	.52	.12	.11
4	1.11	.84	.83	.28	.10	.10	1.08	.69	.67
5	.39	.28	.28	.53	.36	.35	.52	.11	.11
6	.33	.25	.25	.63	.48	.45	.77	.38	.36
7	.28	.22	.21	.76	.62	.59	.71	.32	.30
8	.12	.08	.08	.45	.28	.27	.45	.05	.04

Level 3 LPBH Bounds for an Example Queueing Network

Table 5.12



LPBH Bounds on Class I System Throughput and Mean System Residence Time

Figure 5.2

Time	$6K\|C\|\binom{\|C\|+i}{i}+l(6K\|C\|+K\|C\|^2)\binom{\|C\|+i-1}{i}$
Space	$K\|C\|\binom{\|C\|+i-1}{i}$

The Computational Cost of the Multiple Closed Class LPBH Algorithm

Table 5.13

5.5.5. Algorithm Accuracy

The accuracy of the multiple closed class LPBH algorithm has been studied by applying the algorithm to the queueing networks and population levels used in testing the multiple closed class PBH algorithm. Level 0, 1, 3 and 5 bounds were computed for each queueing network and each population level. Table 5.14 summarizes the results, while tables 5.15 to 5.19 show the effects on the level 3 bounds of changes in the queueing network and population parameters.

Comparing these results to those obtained for the PBH algorithm (tables 4.4 to 4.9), a major conclusion that can be drawn is that the level i LPBH bound is of approximately the same accuracy, and has approximately the same sensitivities to queueing network and population parameters, as the level $i+2$ PBH bound. The one exception to this is that the LPBH bounds are comparatively better, with respect to the PBH bounds, when the interactive workload is large in comparison to the batch workload. (Compare tables 5.17 and 5.18 to tables 4.7 and 4.8.) This exception can be explained by the fact that the expression for $\tilde{n}_{kr}^{(i)}(\vec{N}-\vec{e}_s)$ in the LPBH algorithm is much more accurate when Z_s is large.

Bound Level	Maximal Errors (%)			
	$R_c(\vec{N})$ Aver.	$R_c(\vec{N})$ Max.	$X_c(\vec{N})$ Aver.	$X_c(\vec{N})$ Max.
0	26.7	85.3	16.6	85.3
1	18.4	82.1	9.7	82.1
3	11.1	80.0	5.9	80.0
5	7.7	79.2	4.2	79.2

Maximal Errors for a Range of Bound Levels

Table 5.14

Number of Classes	Maximal Errors (Level 3 Bounds) (%)			
	$R_c(\vec{N})$ Aver.	$R_c(\vec{N})$ Max.	$X_c(\vec{N})$ Aver.	$X_c(\vec{N})$ Max.
2	8.6	61.8	4.5	60.7
4	12.0	69.9	6.5	69.9
6	12.6	80.0	6.7	80.0

Maximal Errors for Various Numbers of Closed Classes

Table 5.15

Number of Fixed-Rate Service Centers	Maximal Errors (Level 3 Bounds) (%)			
	$R_c(\vec{N})$ Aver.	$R_c(\vec{N})$ Max.	$X_c(\vec{N})$ Aver.	$X_c(\vec{N})$ Max.
10	17.2	80.0	9.4	80.0
20	4.9	47.8	2.4	36.0

Maximal Errors for two Different Queueing Network Sizes

Table 5.16

Interactive Workload Intensity	Maximal Errors (Level 3 Bounds) (%)			
	$R_s(\vec{N})$ Aver.	$R_s(\vec{N})$ Max.	$X_s(\vec{N})$ Aver.	$X_s(\vec{N})$ Max.
20	4.4	43.6	1.9	33.5
50	10.5	64.3	5.3	62.4
80	18.3	80.0	10.3	80.0

Maximal Errors for a Range of Total Interactive Workload Intensities

Table 5.17

Batch Workload Intensity	Maximal Errors (Level 3 Bounds) (%)			
	$R_s(\vec{N})$ Aver.	$R_s(\vec{N})$ Max.	$X_s(\vec{N})$ Aver.	$X_s(\vec{N})$ Max.
3	6.1	74.3	3.2	74.3
6	11.2	77.8	5.9	77.8
9	16.0	80.0	8.5	80.0

Maximal Errors for a Range of Total Batch Workload Intensities

Table 5.18

Loading Skewness (value of p)	Maximal Errors (Level 3 Bounds) (%)			
	$R_s(\vec{N})$ Aver.	$R_s(\vec{N})$ Max.	$X_s(\vec{N})$ Aver.	$X_s(\vec{N})$ Max.
0.5	3.2	23.1	1.6	19.9
1.0	8.8	49.0	4.5	46.9
2.0	21.2	80.0	11.5	80.0

Maximal Errors for a Range of p Values

Table 5.19

APPENDIX 5.A

This appendix states and proves lemmas of relevance to chapter five.

Lemma 5.1:

Suppose that two vectors of non-negative mean queue length values, $(n_1^{v(0)}(N), \cdots, n_K^{v(0)}(N))$ and $(n_1^{v'(0)}(N), \cdots, n_K^{v'(0)}(N))$, are such that

$$n_b^{v'(0)}(N) \leq n_b^{v(0)}(N)$$

$$n_k^{v'(0)}(N) \geq n_k^{v(0)}(N) \qquad k \neq b$$

and

$$\sum_{k=1}^{K} n_k^{v(0)}(N) \geq \sum_{k=1}^{K} n_k^{v'(0)}(N)$$

Then for all $i \geq 0$,

$$n_b^{v'(i)}(N) \leq n_b^{v(i)}(N) \tag{5.1-1}$$

$$n_k^{v'(i)}(N) \geq n_k^{v(i)}(N) \qquad k \neq b \tag{5.1-2}$$

and

$$\sum_{k=1}^{K} n_k^{v(i)}(N) \geq \sum_{k=1}^{K} n_k^{v'(i)}(N) \tag{5.1-3}$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by i iterations using the aMVA/PE equations from the values superscripted by $v(0)$ and $v'(0)$, respectively.

Proof:

The proof is by induction on i.

By assumption, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ ($i \geq 0$). Using the aMVA/PE equations, relation (5.1-3) reduces to

$$\sum_{k=1}^{K} \frac{L_k\left(1+n_k^{v(i)}(N)\frac{N-1}{N}\right)}{Z+\sum_{k=1}^{K} L_k\left(1+n_k^{v(i)}(N)\frac{N-1}{N}\right)} N \geq \sum_{k=1}^{K} \frac{L_k\left(1+n_k^{v'(i)}(N)\frac{N-1}{N}\right)}{Z+\sum_{k=1}^{K} L_k\left(1+n_k^{v'(i)}(N)\frac{N-1}{N}\right)} N$$

This last relation holds if

$$\sum_{k=1}^{K} L_k\left(1+n_k^{v(i)}(N)\frac{N-1}{N}\right) \geq \sum_{k=1}^{K} L_k\left(1+n_k^{v'(i)}(N)\frac{N-1}{N}\right)$$

$$\Leftrightarrow \quad L_b\left(n_b^{v(i)}(N)-n_b^{v'(i)}(N)\right) - \sum_{k \neq b} L_k\left(n_k^{v'(i)}(N)-n_k^{v(i)}(N)\right) \geq 0$$

From the inductive hypothesis on relations (5.1-1) and (5.1-2), each term in the above relation is non-negative. This implies that the relation will be established if it can be shown that

$$L_b\left(n_b^{v(i)}(N)-n_b^{v'(i)}(N)\right) - \sum_{k \neq b} L_b\left(n_k^{v'(i)}(N)-n_k^{v(i)}(N)\right) \geq 0$$

$$<==> \quad \sum_{k=1}^{K} n_k^{v(i)}(N) \geq \sum_{k=1}^{K} n_k^{v'(i)}(N)$$

This last relation holds from the inductive hypothesis on relation (5.1-3), establishing relation (5.1-3) for $i+1$.

From relation (5.1-3), relation (5.1-1) or (5.1-2) could be violated for $i+1$ only if there is a service center $j \neq b$ such that

$$n_j^{v(i+1)}(N) \geq n_j^{v'(i+1)}(N)$$

Using the aMVA/PE equations, this relation reduces to

$$\frac{L_j\left(1+n_j^{v(i)}(N)\frac{N-1}{N}\right)}{Z+\sum_{k=1}^{K} L_k\left(1+n_k^{v(i)}(N)\frac{N-1}{N}\right)}N \geq \frac{L_j\left(1+n_j^{v'(i)}(N)\frac{N-1}{N}\right)}{Z+\sum_{k=1}^{K} L_k\left(1+n_k^{v'(i)}(N)\frac{N-1}{N}\right)}N$$

$$<==> \quad \frac{1+n_j^{v(i)}(N)\frac{N-1}{N}}{1+n_j^{v'(i)}(N)\frac{N-1}{N}} \geq \frac{Z+\sum_{k=1}^{K} L_k\left(1+n_k^{v(i)}(N)\frac{N-1}{N}\right)}{Z+\sum_{k=1}^{K} L_k\left(1+n_k^{v'(i)}(N)\frac{N-1}{N}\right)}$$

However, the right-hand side of this last relation has been proved to be greater than or equal to one, while the left-hand side is less than or equal to one by the inductive hypothesis. As this is a contradiction, relations (5.1-1) and (5.1-2) and the lemma are established for $i+1$. By induction, the lemma is established for $i \geq 0$. ∎

Lemma 5.2:

Consider an indexing of the fixed-rate service centers such that $L_k \geq L_{k+1}$, $1 \leq k \leq K-1$. For all $i \geq 0$, $N \geq 1$,

$$n_j^{(i)}{}_{pess}(N-1) \geq n_j^{(i)}{}_{pess}(N)\frac{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N-1)}{\sum_{k=1}^{j} n_k^{(i)}{}_{pess}(N)} \qquad 1 \leq j \leq K \tag{5.2-1}$$

and

$$\frac{N-1}{N}\sum_{k=1}^{K} n_k^{(i)}{}_{pess}(N) \geq \sum_{k=1}^{K} n_k^{(i)}{}_{pess}(N-1) \tag{5.2-2}$$

Proof:

The proof is by induction on i.

For $i=0$, relation (5.2-1) is an equality, while relation (5.2-2) holds from the fact that $b(N) \geq b(N-1)$. This establishes the induction basis. Assume that the lemma holds for i, and consider the lemma for $i+1$ ($i \geq 0$). Since the lemma trivially holds for $N=1$, consider $N+1$ ($N \geq 1$).

Using the PBH equations, relation (5.2-2) reduces to

$$\frac{N}{N+1}\sum_{k=1}^{K} \frac{L_k\left(1+n_k^{(i)}{}_{pess}(N)\right)}{Z+\sum_{k=1}^{K} L_k\left(1+n_k^{(i)}{}_{pess}(N)\right)}(N+1) \geq \sum_{k=1}^{K} \frac{L_k\left(1+n_k^{(i)}{}_{pess}(N-1)\right)}{Z+\sum_{k=1}^{K} L_k\left(1+n_k^{(i)}{}_{pess}(N-1)\right)}N$$

$$<==> \quad \sum_{k=1}^{K} L_k(1+n_k^{(i)}{}_{pess}(N)) \geq \sum_{k=1}^{K} L_k(1+n_k^{(i)}{}_{pess}(N-1))$$

This last relation holds from lemma 3.2 applied to the inductive hypothesis (where n_k is taken as $n_k^{(i)}{}_{pess}(N)$, and $\hat{n}_k$ is taken as $n_k^{(i)}{}_{pess}(N-1)$, for $1 \leq k \leq K$), establishing relation (5.2-2) for $i+1$.

Using the PBH equations, for $1 \leq j \leq K$ relation (5.2-1) reduces to (after canceling common factors from both sides of the relation)

$$1+n_j^{(i)}{}_{pess}(N-1) \geq (1+n_j^{(i)}{}_{pess}(N)) \frac{\sum_{k=1}^{j} L_k(1+n_k^{(i)}{}_{pess}(N-1))}{\sum_{k=1}^{j} L_k(1+n_k^{(i)}{}_{pess}(N))}$$

This last relation holds from lemma 3.2 applied to the inductive hypothesis as before, establishing relation (5.2-1) and the lemma for $i+1$. By induction, the lemma is established for $i \geq 0$. ∎

Lemma 5.3:

Consider an indexing of the fixed-rate service centers such that $L_k \geq L_{k+1}$, $1 \leq k \leq K-1$. For all $N \geq 1$,

$$n_j(N-1) \geq n_j(N) \frac{\sum_{k=1}^{j} n_k(N-1)}{\sum_{k=1}^{j} n_k(N)} \qquad 1 \leq j \leq K \tag{5.3-1}$$

and

$$\frac{N-1}{N} \sum_{k=1}^{K} n_k(N) \geq \sum_{k=1}^{K} n_k(N-1) \tag{5.3-2}$$

Proof:

Relation (5.3-2) follows from Little's equation, and since $R(N) \geq R(N-1)$ for all $N \geq 2$. The proof of relation (5.3-1) is by induction on N. For $N=1$, the relation holds since $n_k(0) = 0$ for all k, establishing the induction basis.

Assume that relation (5.3-1) holds for N ($N \geq 1$), and consider the relation for $N+1$. For $1 \leq j \leq K$,

$$n_j(N) \geq n_j(N+1) \frac{\sum_{k=1}^{j} n_k(N)}{\sum_{k=1}^{j} n_k(N+1)}$$

$$<==> \quad \frac{L_j(1+n_j(N-1))}{Z+R(N)} N \geq \frac{L_j(1+n_j(N))}{Z+R(N+1)} (N+1) \frac{\sum_{k=1}^{j} \frac{L_k(1+n_k(N-1))}{Z+R(N)} N}{\sum_{k=1}^{j} \frac{L_k(1+n_k(N))}{Z+R(N+1)} (N+1)}$$

$$<==> \quad 1+n_j(N-1) \geq (1+n_j(N)) \frac{\sum_{k=1}^{j} L_k(1+n_k(N-1))}{\sum_{k=1}^{j} L_k(1+n_k(N))}$$

which holds from lemma 3.2 applied to the inductive hypothesis and relation (5.3-2), with $\hat{n}_k$ taken as $n_k(N-1)$ and n_k taken as $n_k(N)$ $(1 \le k \le K)$. This establishes relation (5.3-1) for $N+1$. By induction, relation (5.3-1) and the lemma are established for $N \ge 1$. ∎

Lemma 5.4:

For fixed-rate service centers k and j, if $L_k \ge L_j$, then for all $N \ge 1$

$$\frac{L_k^N}{n_k(N)} \ge \frac{L_j^N}{n_j(N)} \tag{5.4-1}$$

Proof:

The proof is by induction on N.

For $N=1$, relation (5.4-1) becomes

$$L_k \frac{1+Z}{L_k} \ge L_j \frac{1+Z}{L_j}$$

which holds with equality, establishing the induction basis. Assume that the lemma holds for N $(N \ge 1)$, and consider the lemma for $N+1$. Multiplying both sides of relation (5.4-1) by $X(N+1)$ and utilizing Little's equation yields

$$\frac{L_k^{N+1}}{R_k(N+1)} \ge \frac{L_j^{N+1}}{R_j(N+1)}$$

Using the MVA mean residence time equation, this last relation reduces to

$$\frac{L_k^N}{1+n_k(N)} \ge \frac{L_j^N}{1+n_j(N)}$$

which holds from the inductive hypothesis and from the fact that $L_k^N \ge L_j^N$. This establishes the lemma for $N+1$, and, by induction, for all $N \ge 1$. ∎

Lemma 5.5:

Consider an indexing of the fixed-rate service centers such that $L_k \ge L_{k+1}$, $1 \le k \le K-1$. For all $N \ge 1$,

$$R_j(N) \ge L_j^N \frac{\sum_{k=1}^{j} R_k(N)}{\sum_{k=1}^{j} L_k^N} \tag{5.5-1}$$

Proof:

Dividing both sides of relation (5.5-1) by $\sum_{k=1}^{j} R_k(N)$, utilizing Little's equation on the left-hand side, and then inverting both sides yields

$$\sum_{k=1}^{j} \frac{n_k(N)}{n_j(N)} \leq \sum_{k=1}^{j} \frac{L_k^N}{L_j^N}$$

which holds from the previous lemma and the assumed indexing of the fixed-rate service centers. ∎

Lemma 5.6:

Given quantities n_k $(1 \leq k \leq K)$, $n_k \geq n_{k+1} \geq 0$ $(1 \leq k \leq K-1)$, $n_1 > 0$, and quantities $\hat{n}_k$ $(1 \leq k \leq K)$, such that

$$\hat{n}_j \geq n_j \frac{\sum_{k=1}^{j} \hat{n}_k}{\sum_{k=1}^{j} n_k} \qquad 1 \leq j \leq K \tag{5.6-1}$$

and

$$\sum_{k=1}^{K} n_k \geq \sum_{k=1}^{K} \hat{n}_k \tag{5.6-2}$$

Then

$$\sum_{k=1}^{j} n_k \geq \sum_{k=1}^{j} \hat{n}_k \frac{\sum_{k=1}^{K} n_k}{\sum_{k=1}^{K} \hat{n}_k} \qquad 1 \leq j \leq K \tag{5.6-3}$$

Proof:

The proof is by downwards induction on j.

For $j=K$, relation (5.6-3) holds with equality, establishing the induction basis. Assume that the lemma holds for $j+1$ $(1 \leq j \leq K-1)$ and consider the lemma for j.

$$\sum_{k=1}^{j} n_k \geq \sum_{k=1}^{j} \hat{n}_k \frac{\sum_{k=1}^{K} n_k}{\sum_{k=1}^{K} \hat{n}_k}$$

$$<==> \quad \sum_{k=1}^{j} n_k \frac{\sum_{k=1}^{j+1} \hat{n}_k}{\sum_{k=1}^{j+1} n_k} \geq \sum_{k=1}^{j} \hat{n}_k \frac{\sum_{k=1}^{K} n_k}{\sum_{k=1}^{K} \hat{n}_k} \frac{\sum_{k=1}^{j+1} \hat{n}_k}{\sum_{k=1}^{j+1} n_k}$$

From the inductive hypothesis, this relation will be shown if it can be shown that

$$\sum_{k=1}^{j} n_k \frac{\sum_{k=1}^{j+1} \hat{n}_k}{\sum_{k=1}^{j+1} n_k} \geq \sum_{k=1}^{j} \hat{n}_k$$

This last relation was established in the proof of relation (3.2-3) in lemma 3.2. This establishes the lemma for j, and, by induction, for $1 \le j \le K$. ∎

Lemma 5.7:

Consider an indexing of the fixed-rate service centers such that $L_k \ge L_{k+1}$, $1 \le k \le K-1$. For all $N \ge 1$,

$$\tilde{n}_j(N) \ge n_j(N) \frac{\sum_{k=1}^{j} \tilde{n}_k(N)}{\sum_{k=1}^{j} n_k(N)} \qquad 1 \le j \le K \tag{5.7-1}$$

and

$$\sum_{k=1}^{K} n_k(N) \ge \sum_{k=1}^{K} \tilde{n}_k(N) \tag{5.7-2}$$

where $\tilde{n}_k(N)$ is defined by

$$\tilde{n}_k(N) = \frac{R_k(N+1) - \frac{L_k^{N+1}}{L^{(N)}}\Delta(N)}{Z + R(N+1) - \frac{L^{(N+1)}}{L^{(N)}}\Delta(N)} N \qquad 1 \le k \le K$$

Proof:

The proof is by induction on N.

Consider $N=1$. For $1 \le j \le K$ relation (5.7-1) becomes (after canceling common factors)

$$L_j\left(1 + \frac{L_j}{1+Z}\right) - L_j^2\left(\frac{1}{1+Z}\right) \ge L_j \frac{\sum_{k=1}^{j} \left(L_k\left(1 + \frac{L_k}{1+Z}\right) - L_k^2\left(\frac{1}{1+Z}\right)\right)}{\sum_{k=1}^{j} L_k}$$

which holds with equality. Relation (5.7-2) becomes

$$\frac{1}{1+Z} \ge \frac{1 + \frac{L^{(2)}}{1+Z} - L^{(2)}\left(\frac{1}{1+Z}\right)}{Z + 1 + \frac{L^{(2)}}{1+Z} - L^{(2)}\left(\frac{1}{1+Z}\right)}$$

which also holds with equality. This establishes the induction basis.

Assume that the lemma holds for $N-1$ ($N \ge 2$), and consider the lemma for N. It will first be established that, for $1 \le j \le K$,

$$n_j(N) - \frac{L_j^N}{L^{(N)}}\Delta(N) \ge n_j(N-1) \frac{\sum_{k=1}^{j} \left(n_k(N) - \frac{L_k^N}{L^{(N)}}\Delta(N)\right)}{\sum_{k=1}^{j} n_k(N-1)} \tag{5.7-3}$$

[Case 1] $\sum_{k=1}^{j} \left(n_k(N) - \frac{L_k^N}{L^{(N)}}\Delta(N)\right) \ge 0$

From the inductive hypothesis on relation (5.7-1), relation (5.7-3) will be shown if it can be shown that

$$n_j(N) - \frac{L_j^N}{L^{(N)}}\Delta(N) \geq \tilde{n}_j(N-1)\frac{\sum_{k=1}^{j}(n_k(N)-\frac{L_k^N}{L^{(N)}}\Delta(N))}{\sum_{k=1}^{j}\tilde{n}_k(N-1)}$$

Canceling common factors on the right-hand side of this relation, and then dividing both sides by $X(N)$ and utilizing Little's equation yields

$$R_j(N) - \frac{L_j^N}{L^{(N)}}\frac{1}{X(N)}\Delta(N) \geq (R_j(N) - \frac{L_j^N}{L^{(N-1)}}\Delta(N-1))\frac{\sum_{k=1}^{j}(R_k(N)-\frac{L_k^N}{L^{(N)}}\frac{1}{X(N)}\Delta(N))}{\sum_{k=1}^{j}(R_k(N)-\frac{L_k^N}{L^{(N-1)}}\Delta(N-1))}$$

$$<==> \quad \frac{1}{L^{(N)}}\frac{1}{X(N)}\Delta(N)(R_j(N)\sum_{k=1}^{j}L_k^N - L_j^N\sum_{k=1}^{j}R_k(N)) \geq$$

$$\frac{1}{L^{(N-1)}}\Delta(N-1)(R_j(N)\sum_{k=1}^{j}L_k^N - L_j^N\sum_{k=1}^{j}R_k(N))$$

This last relation holds since $R_j(N)\sum_{k=1}^{j}L_k^N - L_j^N\sum_{k=1}^{j}R_k(N) \geq 0$ from lemma 5.5, $\frac{1}{L^{(N)}}\frac{1}{X(N)} \geq \frac{1}{L^{(N-1)}}$ from the fact that $\frac{1}{X(N)} \geq L_b$, and $\Delta(N) \geq \Delta(N-1)$ from the fact that throughput is concave (theorem 5.5).

[Case 2] $\sum_{k=1}^{j}(n_k(N)-\frac{L_k^N}{L^{(N)}}\Delta(N)) < 0$

If $n_j(N) - \frac{L_j^N}{L^{(N)}}\Delta(N) \geq 0$, then relation (5.7-3) trivially holds. Assume that $n_j(N) - \frac{L_j^N}{L^{(N)}}\Delta(N) < 0$. Then relation (5.7-3) is equivalent to

$$n_j(N) \leq n_j(N-1)\frac{\sum_{k=1}^{j}n_k(N)\left(\frac{1-\frac{L_k^N}{n_k(N)}\frac{1}{L^{(N)}}\Delta(N)}{1-\frac{L_j^N}{n_j(N)}\frac{1}{L^{(N)}}\Delta(N)}\right)}{\sum_{k=1}^{j}n_k(N-1)}$$

This last relation holds from lemma 5.3, since

$$\frac{1-\frac{L_k^N}{n_k(N)}\frac{1}{L^{(N)}}\Delta(N)}{1-\frac{L_j^N}{n_j(N)}\frac{1}{L^{(N)}}\Delta(N)} \geq 1$$

for $1 \leq k \leq j$ by lemma 5.4 (recalling that $1-\frac{L_j^N}{n_j(N)}\frac{1}{L^{(N)}}\Delta(N)$ is assumed negative). This establishes relation (5.7-3).

Since $\sum_{k=1}^{K}\frac{L_k^N}{L^{(N)}}\Delta(N) = \Delta(N)$, it also follows that

$$\sum_{k=1}^{K} n_k(N-1) = \sum_{k=1}^{K} (n_k(N) - \frac{L_k^N}{L^{(N)}}\Delta(N)) \tag{5.7-4}$$

Noting that relation (5.7-2) for N is equivalent to

$$\sum_{k=1}^{K} L_k(1+n_k(N-1)) \geq \sum_{k=1}^{K} L_k(1+n_k(N) - \frac{L_k^N}{L^{(N)}}\Delta(N))$$

this relation is established by applying lemma 3.2 to relations (5.7-3) and (5.7-4) (with $\hat{n}_k$ taken as $n_k(N) - \frac{L_k^N}{L^{(N)}}\Delta(N)$, and n_k taken as $n_k(N-1)$, $1 \leq k \leq K$). Relation (5.7-1) reduces to (after canceling common factors from both sides of the relation)

$$1+n_j(N) - \frac{L_j^N}{L^{(N)}}\Delta(N) \geq (1+n_j(N-1)) \frac{\sum_{k=1}^{j} L_k(1+n_k(N) - \frac{L_k^N}{L^{(N)}}\Delta(N))}{\sum_{k=1}^{j} L_k(1+n_k(N-1))}$$

which holds from lemma 3.2 applied as before. This establishes the lemma for N, and, by induction, for all $N \geq 1$. ∎

Lemma 5.8:

Consider an indexing of the fixed-rate service centers such that $L_k \geq L_{k+1}$ for $1 \leq k \leq K-1$. For $N=2$,

$$\tilde{n}_j(N-1) \geq n_j(N-1) \frac{\sum_{k=1}^{j} \tilde{n}_k(N-1)}{\sum_{k=1}^{j} n_k(N-1)} \qquad 1 \leq j \leq K \tag{5.8-1}$$

and

$$\sum_{k=1}^{K} n_k(N-1) \geq \sum_{k=1}^{K} \tilde{n}_k(N-1) \tag{5.8-2}$$

where

$$\tilde{n}_k(N-1) = \frac{R_k(N) - (R_k(N+1) - R_k(N))}{Z + R(N) - (R(N+1) - R(N))}(N-1) \qquad 1 \leq k \leq K+1$$

Proof:

Noting that relation (5.8-2) is equivalent to $R(N) - R(N-1) \leq R(N+1) - R(N)$, for $N=2$ this relation becomes

$$(1+L^{(2)}\frac{1}{1+Z}) - 1 \leq (1 + \frac{L^{(2)} + L^{(3)}\frac{1}{1+Z}}{Z+1+L^{(2)}\frac{1}{1+Z}} 2) - (1+L^{(2)}\frac{1}{1+Z})$$

which simplifies to

$$(L^{(2)})^2 \leq L^{(3)}(1+Z)$$

This last relation holds since $(L^{(2)})^2 \leq L^{(3)}$.

Noting that

$$\tilde{n}_j(N-1) \geq n_j(N-1)\frac{\sum_{k=1}^{j}\tilde{n}_k(N-1)}{\sum_{k=1}^{j}n_k(N-1)}$$

reduces to

$$R_j(N)-(R_j(N+1)-R_j(N)) \geq R_j(N-1)\frac{\sum_{k=1}^{j}(R_k(N)-(R_k(N+1)-R_k(N)))}{\sum_{k=1}^{j}R_k(N-1)}$$

for $N=2$ relation (5.8-1) becomes

$$L_j(1+L_j\frac{1}{1+Z})-\left[L_j(1+\frac{L_j+L_j^2\frac{1}{1+Z}}{Z+1+L^{(2)}\frac{1}{1+Z}}2)-L_j(1+L_j\frac{1}{1+Z})\right] \geq$$

$$L_j\frac{\sum_{k=1}^{j}\left[L_k(1+L_k\frac{1}{1+Z})-\left[L_k(1+\frac{L_k+L_k^2\frac{1}{1+Z}}{Z+1+L^{(2)}\frac{1}{1+Z}}2)-L_k(1+L_k\frac{1}{1+Z})\right]\right]}{\sum_{k=1}^{j}L_k}$$

This reduces to

$$(1+Z)(\sum_{k=1}^{j}L_k^3-L_j^2\sum_{k=1}^{j}L_k) \geq L^{(2)}(\sum_{k=1}^{j}L_k^2-L_j\sum_{k=1}^{j}L_k)$$

This relation holds if it holds for $\sum_{k=1}^{j}L_k^3$ minimized. Also, due to the assumed indexing of the fixed-rate service centers, the parenthesized expressions in this relation are non-negative. The relation therefore holds if it holds for Z minimized and $L^{(2)}$ maximized. Substituting in the minimum Z (0), the maximum $L^{(2)}$ $(\sum_{k=1}^{j}L_k^2+L_j^2\frac{1-\sum_{k=1}^{j}L_k}{L_j})$, and the minimum $\sum_{k=1}^{j}L_k^3$ $(\frac{(\sum_{k=1}^{j}L_k^2)^2}{\sum_{k=1}^{j}L_k})$ yields, after simplification,

$$\frac{\sum_{k=1}^{j}L_k^2}{\sum_{k=1}^{j}L_k}(\sum_{k=1}^{j}L_k^2-L_j\sum_{k=1}^{j}L_k) \geq (\sum_{k=1}^{j}L_k^2-L_j\sum_{k=1}^{j}L_k)^2$$

Due to the assumed indexing of the fixed-rate service centers, the parenthesized expression on each side of this last relation is non-negative. The relation therefore holds if

$$\sum_{k=1}^{j}L_k^2 \geq (\sum_{k=1}^{j}L_k)(\sum_{k=1}^{j}L_k^2-L_j\sum_{k=1}^{j}L_k)$$

Since $\sum_{k=1}^{j}L_k \leq 1$, this relation holds, and relation (5.8-1) is established for $N=2$.

This establishes the lemma. ∎

Lemma 5.9:

Consider an indexing of the fixed-rate service centers such that $L_k \geq L_{k+1}$ for $1 \leq k \leq K-1$. For $N \geq 2$,

$$\tilde{n}_j(N-1) \geq n_j(N-1) \frac{\sum_{k=1}^{j} \tilde{n}_k(N-1)}{\sum_{k=1}^{j} n_k(N-1)} \qquad 1 \leq j \leq K \tag{5.9-1}$$

and

$$\sum_{k=1}^{K} n_k(N-1) \geq \sum_{k=1}^{K} \tilde{n}_k(N-1) \tag{5.9-2}$$

where

$$\tilde{n}_k(N-1) = \frac{R_k(N)-(R_k(N+1)-R_k(N))}{Z+R(N)-(R(N+1)-R(N))}(N-1) \qquad 1 \leq k \leq K+1$$

Proof:

The proof is by induction on N.

Relations (5.9-1) and (5.9-2) are shown in lemma 5.8 for $N=2$, establishing the induction basis. Assume that the lemma holds for N ($N \geq 2$), and consider the lemma for $N+1$. It will first be established that, for $1 \leq j \leq K+1$,

$$n_j(N)-(n_j(N+1)-n_j(N)) \geq n_j(N-1) \frac{\sum_{k=1}^{j} (n_k(N)-(n_k(N+1)-n_k(N)))}{\sum_{k=1}^{j} n_k(N-1)} \tag{5.9-3}$$

and

$$\sum_{k=1}^{K} n_k(N-1) \geq \sum_{k=1}^{K} (n_k(N)-(n_k(N+1)-n_k(N))) \tag{5.9-4}$$

Relation (5.9-4) is equivalent to relation (5.9-3) for $j=K+1$. There are two cases to consider in the proof of relation (5.9-3).

[Case 1] $\sum_{k=1}^{j} (n_k(N)-(n_k(N+1)-n_k(N))) \geq 0$

From the inductive hypothesis (noting that relation (5.9-2) yields relation (5.9-1) for $j=K+1$), relation (5.9-3) will be shown if it can be shown that, for $1 \leq j \leq K+1$,

$$n_j(N)-(n_j(N+1)-n_j(N)) \geq \tilde{n}_j(N-1) \frac{\sum_{k=1}^{j} (n_k(N)-(n_k(N+1)-n_k(N)))}{\sum_{k=1}^{j} \tilde{n}_k(N-1)}$$

Substituting the definition of $\tilde{n}_k(N-1)$ ($1 \leq k \leq j$), multiplying both sides by

$\sum_{k=1}^{j}(R_k(N)-(R_k(N+1)-R_k(N)))$, dividing both sides by $X(N)$, and simplifying (with the aid of Little's equation) yields

$$R_j(N)(\frac{X(N+1)}{X(N)}-1)\sum_{k=1}^{j}R_k(N+1) \geq R_j(N+1)(\frac{X(N+1)}{X(N)}-1)\sum_{k=1}^{j}R_k(N)$$

Since $\frac{X(N+1)}{X(N)} \geq 1$, this last relation reduces to

$$R_j(N) \geq R_j(N+1)\frac{\sum_{k=1}^{j}R_k(N)}{\sum_{k=1}^{j}R_k(N+1)} \quad <==> \quad n_j(N) \geq n_j(N+1)\frac{\sum_{k=1}^{j}n_k(N)}{\sum_{k=1}^{j}n_k(N+1)}$$

which holds from lemma 5.3.

[Case 2] $\sum_{k=1}^{j}(n_k(N)-(n_k(N+1)-n_k(N))) < 0$

For this case, $1 \leq j \leq K$, implying that, for $1 \leq k \leq j$,

$$n_k(N)-(n_k(N+1)-n_k(N)) = n_k(N)(2-\frac{n_k(N+1)}{n_k(N)}) = n_k(N)(2-\frac{X(N+1)}{X^{+k}(N)})$$

where $X^{+k}(N)$ denotes the system throughput of the queueing network resulting from the addition of a service center with loading L_k. (This last equality follows from the Convolution algorithm expressions for the system throughput and the mean queue length at a fixed-rate service center.) If $n_j(N)-(n_j(N+1)-n_j(N)) \geq 0$, then relation (5.9-3) trivially holds. Assume that $n_j(N)-(n_j(N+1)-n_j(N)) < 0$. Then relation (5.9-3) is equivalent to

$$n_j(N) \leq n_j(N-1)\frac{\sum_{k=1}^{j}n_k(N)\left(\frac{2-\frac{X(N+1)}{X^{+k}(N)}}{2-\frac{X(N+1)}{X^{+j}(N)}}\right)}{\sum_{k=1}^{j}n_k(N-1)}$$

Since $\frac{2-\frac{X(N+1)}{X^{+k}(N)}}{2-\frac{X(N+1)}{X^{+j}(N)}} \geq 1$ for $1 \leq k \leq j$ due to the assumed indexing of the fixed-rate service centers (recalling that $2-\frac{X(N+1)}{X^{+j}(N)}$ is assumed negative), this last relation holds from lemma 5.3. This establishes relation (5.9-3), and, therefore, relation (5.9-4) as well.

From lemma 3.2 applied to relations (5.9-3) and (5.9-4) (with $\hat{n}_k$ taken as $n_k(N)-(n_k(N+1)-n_k(N))$ and n_k taken as $n_k(N-1)$, $1 \leq k \leq K$),

$$\sum_{k=1}^{K}L_k(1+n_k(N-1)) \geq \sum_{k=1}^{K}L_k(1+n_k(N)-(n_k(N+1)-n_k(N)))$$

$$<==> \quad R(N) \geq R(N+1) - (R(N+2)-R(N+1)) \qquad (5.9\text{-}5)$$

This last relation establishes relation (5.9-2) for $N+1$.

For $N+1$, $1 \le j \le K$, relation (5.9-1) reduces to (after using the definition of $\tilde{n}_k(N)$, Little's equation, and the MVA mean residence time equation)

$$1+n_j(N)-(n_j(N+1)-n_j(N)) \ge (1+n_j(N-1)) \frac{\sum_{k=1}^{j} L_k(1+n_k(N)-(n_k(N+1)-n_k(N)))}{\sum_{k=1}^{j} L_k(1+n_k(N-1))}$$

This last relation holds from lemma 3.2 applied as before, establishing relation (5.9-1) for $N+1$. This establishes the lemma for $N+1$, and, by induction, for all $N \ge 2$. ∎

Lemma 5.10:

Suppose that two mean queue length allocations with associated pessimistic and optimistic heaps, $[n_{ks}^{v(0)}(\vec{N})]_{1\le k\le K, s\in C}$ $(0 \le n_{ks}^{v(0)}(\vec{N}) \le n_{ks}(\vec{N}))$ with $n_s^{hv(0)}{}_{pess}(\vec{N})$, $n_s^{hv(0)}{}_{opt}(\vec{N})$ (for all $s \in C$), and $[n_{ks}^{v'(0)}(\vec{N})]_{1\le k\le K, s\in C}$ $(0 \le n_{ks}^{v'(0)}(\vec{N}) \le n_{ks}(\vec{N}))$ with $n_s^{hv'(0)}{}_{pess}(\vec{N})$, $n_s^{hv'(0)}{}_{opt}(\vec{N})$ (for all $s \in C$), are such that, for each $s \in C$,

$$n_{ks}^{v(0)}(\vec{N}) \le n_{ks}^{v'(0)}(\vec{N}) \qquad 1 \le k \le K$$

$$n_s^{hv(0)}{}_{pess}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v(0)}(\vec{N}) \ge n_s^{hv'(0)}{}_{pess}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v'(0)}(\vec{N})$$

and

$$n_s^{hv(0)}{}_{opt}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v(0)}(\vec{N}) \le n_s^{hv'(0)}{}_{opt}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v'(0)}(\vec{N})$$

Then for $s \in C$ and $i \ge 0$,

$$n_{ks}^{v(i)}(\vec{N}) \le n_{ks}^{v'(i)}(\vec{N}) \qquad 1 \le k \le K \tag{5.10-1}$$

$$n_s^{hv(i)}{}_{pess}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v(i)}(\vec{N}) \ge n_s^{hv'(i)}{}_{pess}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v'(i)}(\vec{N}) \tag{5.10-2}$$

and

$$n_s^{hv(i)}{}_{opt}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v(i)}(\vec{N}) \le n_s^{hv'(i)}{}_{opt}(\vec{N}) + \sum_{k=1}^{K} n_{ks}^{v'(i)}(\vec{N}) \tag{5.10-3}$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by i iterations using the equations of the multiple class looping algorithm from the values superscripted by $v(0)$ and $v'(0)$, respectively.

Proof:

The proof is by induction on i.

By assumption, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ $(i \ge 0)$. Relation (5.10-2) will be established if it can be shown that, for each $s \in C$,

$$R_s^{v(i+1)}{}_{pess}(\vec{N}) \ge R_s^{v'(i+1)}{}_{pess}(\vec{N}) \tag{5.10-4}$$

which reduces to, using equation (5-44),

$$\sum_{k=1}^{K} R_{ks}^{v(i+1)}(\vec{N}) + \sum_{r\in C} L_{sr}^{h}{}_{VB} n_r^{hv(i)}{}_{pess}(\vec{N}-\vec{e}_s) \ge \sum_{k=1}^{K} R_{ks}^{v'(i+1)}(\vec{N}) + \sum_{r\in C} L_{sr}^{h}{}_{VB} n_r^{hv'(i)}{}_{pess}(\vec{N}-\vec{e}_s)$$

From equation (5-42), this relation can be expanded to yield, after regrouping terms and canceling $\sum_{k=1}^{K} L_{ks}$ from both sides of the relation,

$$\sum_{r \in C} L^{h}_{sr\ UB}(n^{hv(i)}_{r\ pess}(\vec{N}-\vec{e}_s) - n^{hv'(i)}_{r\ pess}(\vec{N}-\vec{e}_s)) \geq$$

$$\sum_{r \in C} \sum_{k=1}^{K} L_{ks}(\tilde{n}^{v'(i)}_{kr}(\vec{N}-\vec{e}_s) - \tilde{n}^{v(i)}_{kr}(\vec{N}-\vec{e}_s)) \qquad (5.10\text{-}5)$$

From the inductive hypothesis on relation (5.10-3), $R^{v'(i)}_{s\ opt}(\vec{N}) \geq R^{v(i)}_{s\ opt}(\vec{N})$. In conjunction with equation (5-46) and the inductive hypothesis on relation (5.10-1), this yields

$$\tilde{n}^{v'(i)}_{kr}(\vec{N}-\vec{e}_s) \geq \tilde{n}^{v(i)}_{kr}(\vec{N}-\vec{e}_s) \qquad (5.10\text{-}6)$$

implying that each term in the double summation on the right-hand side of relation (5.10-5) is non-negative. These terms represent an allocation of a heap for each class r, given by $\sum_{k=1}^{K} \tilde{n}^{v'(i)}_{kr}(\vec{N}-\vec{e}_s) - \sum_{k=1}^{K} \tilde{n}^{v(i)}_{kr}(\vec{N}-\vec{e}_s)$, to the fixed-rate service centers. Since, by construction, $L^{h}_{sr\ UB}$ is an upper bound on the mean class s loading when encountering customers from the class r heap, the relation holds if it holds with L_{ks} replaced by $L^{h}_{sr\ UB}$. Making this substitution, and applying equation (5-48), results in an equality. This establishes relation (5.10-2) for $i+1$.

From relation (5.10-6) and equations (5-42) and (5-45), relation (5.10-4) also establishes relation (5.10-1) for $i+1$. Relation (5.10-3) will be established for $i+1$ if it can be shown that, for each $s \in C$,

$$R^{v(i+1)}_{s\ opt}(\vec{N}) \leq R^{v'(i+1)}_{s\ opt}(\vec{N}) \qquad (5.10\text{-}7)$$

From relation (5.10-4), $a^{v'(i+1)}_{s}(\vec{N}) \geq a^{v(i+1)}_{s}(\vec{N})$. Therefore, from equation (5-43), relation (5.10-7) will be shown if it can be shown that

$$\sum_{k=1}^{K} R^{v(i+1)}_{ks}(\vec{N}) + \sum_{r \in C} L^{h}_{sr\ LB} n^{hv(i)}_{r\ opt}(\vec{N}-\vec{e}_s) \leq \sum_{k=1}^{K} R^{v'(i+1)}_{ks}(\vec{N}) + \sum_{r \in C} L^{h}_{sr\ LB} n^{hv'(i)}_{r\ opt}(\vec{N}-\vec{e}_s)$$

Using equation (5-42), this relation can be expanded to yield, after regrouping terms and canceling $\sum_{k=1}^{K} L_{ks}$ from both sides of the relation,

$$\sum_{r \in C} L^{h}_{sr\ LB}(n^{hv(i)}_{r\ opt}(\vec{N}-\vec{e}_s) - n^{hv'(i)}_{r\ opt}(\vec{N}-\vec{e}_s)) \leq \sum_{r \in C} \sum_{k=1}^{K} L_{ks}(\tilde{n}^{v'(i)}_{kr}(\vec{N}-\vec{e}_s) - \tilde{n}^{v(i)}_{kr}(\vec{N}-\vec{e}_s))$$

From relation (5.10-6), each term in the double summation on the right-hand side of this last relation is non-negative. These terms represent an allocation of a heap for each class r, given by $\sum_{k=1}^{K} \tilde{n}^{v'(i)}_{kr}(\vec{N}-\vec{e}_s) - \sum_{k=1}^{K} \tilde{n}^{v(i)}_{kr}(\vec{N}-\vec{e}_s)$, to the fixed-rate service centers. Since, by construction, $L^{h}_{sr\ LB}$ is a lower bound on the mean class s loading when encountering customers from the class r heap, the relation holds if it holds with L_{ks} replaced by $L^{h}_{sr\ LB}$. Making this substitution, and applying equation (5-47), results in an equality. This establishes the lemma for $i+1$, and, by induction, for all $i \geq 0$. ∎

APPENDIX 5.B

This appendix contains an analysis of the single class aMVA/PE algorithm, given in section 5.1, for initial performance measure estimates corresponding to the optimistic assumption that the mean residence time at each service center is equal to the service center loading. In particular, it is proved that these initial estimates guarantee monotonic convergence to the solution of the aMVA/PE equations, and that convergence is from the optimistic side of this solution.

To establish this claim, it is sufficient to prove that the mean queue length values produced after i iterations are majorized by those produced after $i+1$ iterations ($i \geq 0$). This would show, in particular, that $R^{(i+1)}(N) \geq R^{(i)}(N)$, which is sufficient to imply convergence since $R^{(i)}(N)$ is bounded above by $1+(N-1)L_b$ for all $i \geq 0$.

Given initial estimates

$$R^{(0)}(N) = 1$$

$$R_k^{(0)}(N) = L_k \qquad 1 \leq k \leq K$$

and, from Little's equation,

$$n_k^{(0)}(N) = \frac{L_k}{Z+1} N \qquad 1 \leq k \leq K$$

it is easily verified that

$$n_j^{(0)}(N) \geq n_j^{(1)}(N) \frac{\sum_{k=1}^{j} n_k^{(0)}(N)}{\sum_{k=1}^{j} n_k^{(1)}(N)} \qquad 1 \leq j \leq K$$

and

$$\sum_{k=1}^{K} n_k^{(1)}(N) \geq \sum_{k=1}^{K} n_k^{(0)}(N)$$

Therefore, it only remains to show that majorization is preserved by the aMVA/PE equations.

To this end, consider an indexing of the fixed-rate service centers such that $L_k \geq L_{k+1}$, $1 \leq k \leq K-1$. Suppose that two vectors of non-negative mean queue length values, $(n_1^{\nu(0)}(N), \cdots, n_K^{\nu(0)}(N))$ and $(n_1^{\nu'(0)}(N), \cdots, n_K^{\nu'(0)}(N))$, are such that

$$n_j^{\nu'(0)}(N) \geq n_j^{\nu(0)}(N) \frac{\sum_{k=1}^{j} n_k^{\nu'(0)}(N)}{\sum_{k=1}^{j} n_k^{\nu(0)}(N)} \qquad 1 \leq j \leq K$$

and

$$\sum_{k=1}^{K} n_k^{\nu(0)}(N) \geq \sum_{k=1}^{K} n_k^{\nu'(0)}(N)$$

It is then claimed that, for all $i \geq 0$,

$$n_j^{\nu'(i)}(N) \geq n_j^{\nu(i)}(N) \frac{\sum_{k=1}^{j} n_k^{\nu'(i)}(N)}{\sum_{k=1}^{j} n_k^{\nu(i)}(N)} \qquad 1 \leq j \leq K \qquad \text{(5.B-1)}$$

and

$$\sum_{k=1}^{K} n_k^{v(i)}(N) \geq \sum_{k=1}^{K} n_k^{v'(i)}(N) \qquad \text{(5.B-2)}$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by i iterations using the aMVA/PE equations from the values superscripted by $v(0)$ and $v'(0)$, respectively.

The proof is by induction on i.

By assumption, the claim holds for the induction basis $i=0$. Assume that the claim holds for i, and consider the claim for $i+1$ ($i \geq 0$). Using the aMVA/PE equations, relation (5.B-2) reduces to

$$\frac{\sum_{k=1}^{K} L_k\left(1+n_k^{v(i)}(N)\frac{N-1}{N}\right)}{Z+\sum_{k=1}^{K} L_k\left(1+n_k^{v(i)}(N)\frac{N-1}{N}\right)}N \geq \frac{\sum_{k=1}^{K} L_k\left(1+n_k^{v'(i)}(N)\frac{N-1}{N}\right)}{Z+\sum_{k=1}^{K} L_k\left(1+n_k^{v'(i)}(N)\frac{N-1}{N}\right)}N$$

$$<==> \quad \sum_{k=1}^{K} L_k\left(1+n_k^{v(i)}(N)\frac{N-1}{N}\right) \geq \sum_{k=1}^{K} L_k\left(1+n_k^{v'(i)}(N)\frac{N-1}{N}\right)$$

This last relation holds from lemma 3.2 applied to the inductive hypothesis, establishing relation (5.B-2) for $i+1$.

Using the aMVA/PE equations, for $1 \leq j \leq K$ relation (5.B-1) reduces to (after canceling common factors from both sides of the relation)

$$1+n_j^{v'(i)}(N)\frac{N-1}{N} \geq \left(1+n_j^{v(i)}(N)\frac{N-1}{N}\right)\frac{\sum_{k=1}^{j} L_k\left(1+n_k^{v'(i)}(N)\frac{N-1}{N}\right)}{\sum_{k=1}^{j} L_k\left(1+n_k^{v(i)}(N)\frac{N-1}{N}\right)}$$

This last relation holds from lemma 3.2 applied as before, establishing relation (5.B-1) and the claim for $i+1$. By induction, the claim is established for $i \geq 0$.

CHAPTER 6

BOUNDING ALGORITHMS BASED ON CONVOLUTION

The PBH and LPBH algorithms were derived by applying the methodology presented in chapter two to the MVA and symbolic MVA solution algorithms. This methodology can also be applied to the other solution algorithms that are available for the analysis of separable queueing networks. This chapter treats perhaps the most conceptually important of these, the Convolution algorithm. (Other solution algorithms can, in general, be viewed as modifications of the Convolution or MVA algorithms.)

For queueing networks with only fixed-rate and delay service centers, the Convolution algorithm can be formulated as a customer recursion method, in which the algorithm recursion is based on using the solutions of the networks with one fewer customer (one network for each customer class) to compute the solution of a more complex network. Section one of this chapter considers the customer recursion formulation of Convolution; this formulation is found to be unsuitable for use as a bounding algorithm basis. A more suitable formulation of Convolution is based on service center recursion, in which the solution of a network is computed from the solution of a simpler network with one fewer service center. Section two of this chapter considers bounding algorithms based on the service center recursion formulation of Convolution. Lemmas required in the body of the chapter are proved in appendix 6.A.

6.1. Bounding Algorithms Based on Customer Recursion Convolution

6.1.1. Customer Recursion Convolution

When applied to closed queueing networks, the Convolution solution algorithm computes the elements of a $|C|+1$ dimensional array G of normalization constants. For queueing networks with fixed-rate and delay service centers only, it is best to index the aggregate delay service center (if none, Z can be taken as 0 in the following development) as service center 0. With this indexing, $G(\vec{n},k)$ refers to the population level $\vec{n}$ normalization constant for the queueing network consisting of service centers 0 through k.

To obtain primary performance measures for population level $\vec{N}$, it is necessary to obtain $G(\vec{N},K)$ and $G(\vec{N}-\vec{e}_s,K)$ for all $s \in C$; these values are obtained by recursively computing elements of G using, for $1 \le k \le K$, $\vec{n} \ge 0$,

$$G(\vec{n},k) = G(\vec{n},k-1) + \sum_{r \in C} L_{kr}\, G(\vec{n}-\vec{e}_r,k) \tag{6-1}$$

(where $G(\vec{n}-\vec{e}_r,k)$ is taken as zero if $n_r=0$), and

$$G(\vec{n},0) = \prod_{r \in C} \frac{Z_r^{n_r}}{n_r!} \tag{6-2}$$

The order of computation of the elements of G determines whether the algorithm uses customer recursion or service center recursion. In a step of customer recursion Convolution, $G(\vec{n},k)$ for all k is computed from all $G(\vec{n}-\vec{e}_r,j)$ quantities ($r \in C$, $0 \le j \le K$).

The unsuitability of customer recursion Convolution as a bounding algorithm basis is best shown by considering its equations for queueing networks without delay service centers. For these queueing networks, $G(\vec{n},0)$ for $\vec{n} \ne 0$ is taken as 0, making the resulting equations completely independent of the population level $\vec{N}$ for which performance measures are desired. This independence has important consequences for any bounding algorithm based on customer

recursion Convolution. Assuming that the Convolution equations are utilized unmodified in such a bounding algorithm, the level i normalization constants for population level $\vec{N}$ must differ from those for population level $\vec{M}$ (for $\vec{N}$ and $\vec{M}$ such that $i \leq M_r$ and $i \leq N_r$ for all $r \in C$), only due to differing initial estimates. This implies, for example, that the system throughput estimates for class s, $\frac{G^{(i-1)}(\vec{M}-\vec{e}_s,k)}{G^{(i)}(\vec{M},k)}$ and $\frac{G^{(i-1)}(\vec{N}-\vec{e}_s,k)}{G^{(i)}(\vec{N},k)}$, must differ only due to differing initial estimates, implying that only very good initial estimates will yield acceptable accuracy. This is in marked contrast to the PBH algorithm.

To illustrate these disadvantages, a bounding algorithm based on customer recursion Convolution is developed in the next subsection for closed single class queueing networks with fixed-rate service centers only. Only bounds on primary performance measures will be considered.

6.1.2. An Example Algorithm

For the type of queueing network being treated, equation (6-1) reduces to, for $n \geq 1$, $1 \leq k \leq K$,

$$G(n,k) = G(n,k-1) + L_k G(n-1,k) \tag{6-3}$$

where $G(n,0) = 0$ for $n>0$. To obtain primary performance measures, this equation is used to compute $G(N,K)$ and $G(N-1,K)$, starting from $G(0,k) = 1$ for all k. The example bounding algorithm based on customer recursion Convolution utilizes equation (6-3) unchanged; bound level i normalization constants ($i \geq 1$) for population level N are defined by

$$G^{(i)}(N,k) = G^{(i)}(N,k-1) + L_k G^{(i-1)}(N-1,k) \tag{6-4}$$

where $G^{(i)}(N,0) = 0$ for $N>0$.

As in the single closed class PBH algorithm, two computations are required to yield both "optimistic" and "pessimistic" quantities. A vector of positive normalization constant values $(G^v(N,1), \cdots, G^v(N,K))$ is optimistic with respect to

another vector $(G^{v'}(N,1), \cdots, G^{v'}(N,K))$ if

$$\frac{G^{v}(N,k)}{G^{v}(N,j)} \leq \frac{G^{v'}(N,k)}{G^{v'}(N,j)}$$

for all k, j such that $1 \leq k \leq j \leq K$. It is proved in lemma 6.1 in appendix 6.A that the optimistic or pessimistic character of a vector of normalization constants is preserved by the customer recursion Convolution equations.

In the bounding algorithm optimistic computation, the service centers are indexed so that $L_k \geq L_{k+1}$ for $1 \leq k \leq K-1$. (In general, the performance measures produced are dependent on the service center ordering used.) One choice of optimistic initial estimates suitable for this ordering is

$$G_{opt}^{(0)}(N,k) = \binom{N+k-1}{k-1} \qquad 1 \leq k \leq K \tag{6-5}$$

The following theorem uses lemma 6.1 to establish a relationship between level i and level $i+1$ normalization constants computed during the optimistic computation.

Theorem 6.1:

For $N-1 \geq i \geq 0$, $1 \leq k \leq j \leq K$,

$$\frac{G_{opt}^{(i)}(N,k)}{G_{opt}^{(i)}(N,j)} \leq \frac{G_{opt}^{(i+1)}(N,k)}{G_{opt}^{(i+1)}(N,j)}$$

Proof:

From lemma 6.1, it need only be established that

$$\frac{G_{opt}^{(0)}(N-i,k)}{G_{opt}^{(0)}(N-i,j)} \leq \frac{G_{opt}^{(1)}(N-i,k)}{G_{opt}^{(1)}(N-i,j)}$$

for $1 \leq k \leq j \leq K$. From equations (6-4) and (6-5), this relation is equivalent to

$$\frac{\binom{N-i+k-1}{k-1}}{\binom{N-i+j-1}{j-1}} \leq \frac{\sum_{l=1}^{k} L_l \binom{N-i+l-2}{l-1}}{\sum_{l=1}^{j} L_l \binom{N-i+l-2}{l-1}} \tag{6-6}$$

Noting that the right-hand side of relation (6-6) is equivalent to

$$\frac{1}{1+\dfrac{\sum_{l_1=k+1}^{j} L_{l_1}\binom{N-i+l_1-2}{l_1-1}}{\sum_{l_2=1}^{k} L_{l_2}\binom{N-i+l_2-2}{l_2-1}}}$$

the service center indexing used in the optimistic computation implies that relation (6-6) will be shown if it can be shown for $L_l = L_k$ for $1 \le l \le j$. Making this substitution and canceling L_k yields

$$\frac{\binom{N-i+k-1}{k-1}}{\binom{N-i+j-1}{j-1}} \le \frac{\sum_{l=1}^{k}\binom{N-i+l-2}{l-1}}{\sum_{l=1}^{j}\binom{N-i+l-2}{l-1}}$$

$$<==> \quad \frac{\binom{N-i+k-1}{k-1}}{\sum_{l=1}^{k}\binom{N-i+l-2}{l-1}} \le \frac{\binom{N-i+j-1}{j-1}}{\sum_{l=1}^{j}\binom{N-i+l-2}{l-1}}$$

This last relation holds with equality since both the left-hand and the right-hand sides are equal to 1. ∎

In the bounding algorithm pessimistic computation, the service centers are indexed so that $L_k \le L_{k+1}$ for $1 \le k \le K-1$. One choice of pessimistic initial estimates suitable for this ordering is the same as the choice of optimistic initial estimates:

$$G_{pess}^{(0)}(N,k) = \binom{N+k-1}{k-1} \qquad 1 \le k \le K \tag{6-7}$$

The following theorem uses lemma 6.1 to establish a relationship between level i and level $i+1$ normalization constants computed during the pessimistic computation.

Theorem 6.2:

For $N-1 \ge i \ge 0$, $1 \le k \le j \le K$,

$$\frac{G_{pess}^{(i+1)}(N,k)}{G_{pess}^{(i+1)}(N,j)} \le \frac{G_{pess}^{(i)}(N,k)}{G_{pess}^{(i)}(N,j)}$$

Proof:

From lemma 6.1, it need only be established that

$$\frac{G_{pess}^{(1)}(N-i,k)}{G_{pess}^{(1)}(N-i,j)} \leq \frac{G_{pess}^{(0)}(N-i,k)}{G_{pess}^{(0)}(N-i,j)}$$

for $1 \leq k \leq j \leq K$. From equations (6-4) and (6-7), this relation is equivalent to

$$\frac{\sum_{l=1}^{k} L_l \binom{N-i+l-2}{l-1}}{\sum_{l=1}^{j} L_l \binom{N-i+l-2}{l-1}} \leq \frac{\binom{N-i+k-1}{k-1}}{\binom{N-i+j-1}{j-1}} \qquad (6\text{-}8)$$

Noting that the left-hand side of relation (6-8) is equivalent to

$$\frac{1}{1+\dfrac{\sum_{l_1=k+1}^{j} L_{l_1} \binom{N-i+l_1-2}{l_1-1}}{\sum_{l_2=1}^{k} L_{l_2} \binom{N-i+l_2-2}{l_2-1}}}$$

the service center indexing used in the pessimistic computation implies that relation (6-8) will be shown if it can be shown for $L_l = L_k$ for $1 \leq l \leq j$. The resulting relation was shown to hold with equality in the proof of theorem 6.1. ∎

In the next theorem, theorems 6.1 and 6.2 are used to construct primary performance measure bounds from the normalization constants yielded during the optimistic and pessimistic computations. The superscripts *UB* and *LB* denote upper and lower bounds, respectively, as obtained from the bounding algorithm as detailed in the theorem. The level i bounds ($i \geq 1$) are those derived from the level i and level $i-1$ normalization constants; level 0 bounds are not defined.

Theorem 6.3:

For $N \geq i \geq 1$,

$$N\frac{G_{pess}^{(i)}(N,K)}{G_{pess}^{(i-1)}(N-1,K)} \geq R(N) \geq max\left[N\frac{G_{opt}^{(i)}(N,K)}{G_{opt}^{(i-1)}(N-1,K)}, a(N)\right] \qquad (6\text{-}9)$$

$$\frac{N}{R_{LB}^{(i)}(N)} \geq X(N) \geq \frac{N}{R_{UB}^{(i)}(N)} \tag{6-10}$$

and

$$X_{UB}^{(i)}(N)L_k \geq U_k(N) \geq X_{LB}^{(i)}(N)L_k \tag{6-11}$$

Proof:

The bound of $\alpha(N)$ in relation (6-9) follows from the correctness of ABA. Relations (6-10) and (6-11) follow from relation (6-9) by Little's equation. Since

$$R(N) = N\frac{G(N,K)}{G(N-1,K)}$$

from Little's equation and the Convolution solution algorithm expression for $X(N)$, relation (6-9) and the theorem will be shown if it can be shown that

$$\frac{G_{pess}^{(i)}(N,K)}{G_{pess}^{(i-1)}(N-1,K)} \geq \frac{G(N,K)}{G(N-1,K)} \geq \frac{G_{opt}^{(i)}(N,K)}{G_{opt}^{(i-1)}(N-1,K)}$$

$$<==> \sum_{k=1}^{K} L_k \frac{G_{pess}^{(i-1)}(N-1,k)}{G_{pess}^{(i-1)}(N-1,K)} \geq \sum_{k=1}^{K} L_k \frac{G(N-1,k)}{G(N-1,K)} \geq \sum_{k=1}^{K} L_k \frac{G_{opt}^{(i-1)}(N-1,k)}{G_{opt}^{(i-1)}(N-1,K)}$$

This last relation holds from theorems 6.1 and 6.2, noting that $G^{(N)}(N,k)$ is exact for all k and that therefore, by transitivity, the level $i+1$ normalization constants in these theorems can be replaced by the exact normalization constants. ∎

Table 6.2 compares bounds yielded by the customer recursion Convolution bounding algorithm to those yielded by the PBH algorithm, for the example queueing network of table 6.1 and a population of 10 customers. Note that the bounds yielded by the former algorithm are quite poor. Similar results have been obtained with other examples.

Service Center Loadings														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
.15	.13	.13	.10	.10	.08	.06	.05	.05	.04	.04	.03	.02	.01	.01

An Example Queucing Network

Table 6.1

Bounding Algorithm Based on Customer Iteration Convolution				
Bound Level	Bounds on $R(10)$		Bounds on $X(10)$	
	Upper	Lower	Upper	Lower
1	3.15	1.50	6.66	3.17
2	2.85	1.50	6.66	3.51
4	2.45	1.50	6.66	4.08
6	2.20	1.76	5.68	4.53

PBH Algorithm				
Bound Level	Bounds on $R(10)$		Bounds on $X(10)$	
	Upper	Lower	Upper	Lower
1	2.35	1.60	6.25	4.26
2	2.13	1.86	5.36	4.70
4	2.00	1.95	5.12	4.99
6	1.97	1.96	5.09	5.07

A Comparison Between PBH Bounds and Bounds Yielded by an Algorithm Based on Customer Iteration Convolution

Table 6.2

6.2. Bounding Algorithms Based on Service Center Recursion Convolution

6.2.1. Background

Service center recursion Convolution can be applied to queueing networks containing load-dependent service centers, unlike customer recursion Convolution. For these more general queueing networks, the aggregate delay service center, if present, will be indexed as service center $K+1$, as in previous chapters. $G(\vec{n},k)$ $(1 \le k \le K+1)$ then refers to the population level $\vec{n}$ normalization constant for the queueing network consisting of service centers 1 through k. To obtain primary performance measures for population level $\vec{N}$, it is necessary to obtain $G(\vec{N},K+1)$ and $G(\vec{N}-\vec{e}_s,K+1)$ for all $s \in C$, if there is an aggregate delay service center, and $G(\vec{N},K)$ and $G(\vec{N}-\vec{e}_s,K)$ for all $s \in C$, otherwise.

In a step of service center recursion Convolution, $G(\vec{n},k)$ for all $\vec{n}$ such that $0 \le \vec{n} \le \vec{N}$ is computed from all $G(\vec{m},k-1)$. This is done using equation (6-1) for a fixed-rate service center k,

$$G(\vec{n},k) = \sum_{\vec{m} \le \vec{n}} G(\vec{n}-\vec{m},k-1) \prod_{r \in C} \frac{\prod_{l=prec_r+1}^{prec_r+m_r} L_{kr}(l)}{m_r!} \qquad (6\text{-}12)$$

for a load-dependent service center k at which the service rate is dependent only on the total number of customers present (where $prec_r$ is defined as in subsection 2.2.3), and

$$G(\vec{n},K+1) = \sum_{\vec{m} \le \vec{n}} G(\vec{n}-\vec{m},K) \prod_{r \in C} \frac{Z_r^{m_r}}{m_r!} \qquad (6\text{-}13)$$

for an aggregate delay service center. $G(\vec{n},0)$ is taken as 0 for all $\vec{n} \neq 0$, while $G(\vec{0},0)$ is taken as 1.

A bounding algorithm based on service center recursion Convolution selects some set $\{k_1, \cdots, k_i\}$ of the queueing network service centers $(1 \le k_j \le K+1$ for $1 \le j \le i)$. Normalization constant estimates are chosen for the

queueing network formed by the remaining service centers, for all populations $\vec{n} \leq \vec{N}$. Equations (6-1), (6-12) and (6-13) (or possibly modifications of these) are then applied to the queueing network formed by the service centers in the chosen set, with $G(\vec{n},k_0)$ taken as the population level $\vec{n}$ normalization constant estimate for the network containing the remaining service centers. The level i of the resulting bounds is equal to the number of recursive steps that are required, which equals the number of service centers in the chosen set.

For queueing networks with multiple customer classes, however, even a single step of service center recursion may be computationally prohibitive. The dominant term in the computational cost of a step involving a fixed-rate service center includes a factor of $\prod_{r \in C}(N_r+1)$, which for a load-dependent service center becomes $\prod_{r \in C}(N_r+1)(N_r+2)$. For this reason, the example bounding algorithm of the next subsection will be developed only for queueing networks with a single closed class. For clarity, only queueing networks with no open classes will be treated, and only bounds on the primary performance measures (including utilizations only for fixed-rate service centers) will be derived.

6.2.2. An Example Algorithm

The example bounding algorithm based on service center recursion Convolution, when applied to the type of queueing network being treated, utilizes equations (6-1), (6-12) and (6-13) unchanged:

$$G^{(i)}(n,k_i) = G^{(i-1)}(n,k_{i-1}) + L_{k_i} G^{(i)}(n-1,k_i) \tag{6-14}$$

for a fixed-rate service center k_i,

$$G^{(i)}(n,k_i) = \sum_{m=0}^{n} G^{(i-1)}(n-m,k_{i-1}) \frac{\prod_{l=1}^{m} L_{k_i}(l)}{m!} \tag{6-15}$$

for a load-dependent service center k_i, and

$$G^{(i)}(n,k_i) = \sum_{m=0}^{n} G^{(i-1)}(n-m,k_{i-1}) \frac{Z^m}{m!} \tag{6-16}$$

for an aggregate delay service center k_i. These equations are applied to the queueing network formed by the chosen set of service centers $\{k_1, \cdots, k_i\}$.

As with the single closed class PBH algorithm, two computations are required: one from "optimistic" $G^{(0)}(n,k_0)$ values, and one from "pessimistic" $G^{(0)}(n,k_0)$ values. Positive normalization constant values $G^v(n,k)$ are optimistic with respect to $G^{v'}(n,k)$ values if $G^v(0,k) = G^{v'}(0,k) = 1$, and if, for each n such that $1 \leq n \leq N$,

$$\frac{G^v(n-1,k)}{G^v(n,k)} \geq \frac{G^{v'}(n-1,k)}{G^{v'}(n,k)}$$

In lemma 6.2 in appendix 6.A, it is proved that the optimistic or pessimistic character of initial normalization constant estimates is preserved by the service center recursion Convolution equations. This lemma requires that $\frac{L_j(m)}{m} \geq \frac{L_j(m+1)}{m+1}$ hold for $1 \leq m \leq N-1$ and each load-dependent service center j; this will be assumed in the remainder of this section. (As noted in the section treating load dependence in chapter four, this assumption corresponds to a non-decreasing total service rate and is usually satisfied in the computer system modelling context.)

The $G^{(0)}(n,k_0)$ values are estimates of the normalization constants for the queueing network formed by the service centers not included in $\{k_1, \cdots, k_i\}$. Optimistic $G^{(0)}(n,k_0)$ values can be obtained from

$$G^{(0)}_{opt}(n,k_0) = \frac{1}{\prod_{m=1}^{n} X_{opt}^{(-k_1)\cdots(-k_i)}(m)}$$

where $X_{opt}^{(-k_1)\cdots(-k_i)}(m)$ denotes an optimistic bound on the system throughput of this queueing network with m customers. Similarly, pessimistic $G^{(0)}(n,k_0)$ values can be obtained from

$$G^{(0)}_{pess}(n,k_0) = \frac{1}{\prod_{m=1}^{n} X_{pess}^{(-k_1)\cdots(-k_i)}(m)}$$

The symbolic formulation of the PBH algorithm can be used to obtain all the necessary system throughput bounds (assuming that all load-dependent service centers are included in the set $\{k_1, \cdots, k_i\}$) in time $c_1 N + c_2(K-i)$, for constants c_1 and c_2 that depend on the PBH bound level (see table 3.6).

The following theorem uses lemma 6.2 to construct primary performance measure bounds from the normalization constants yielded during the optimistic and pessimistic computations. The superscripts *UB* and *LB* denote upper and lower bounds, respectively, as obtained from the bounding algorithm as detailed in the theorem.

Theorem 6.4:

For $K \geq i \geq 0$, $N \geq 1$,

$$N\frac{G_{pess}^{(i)}(N,k_i)}{G_{pess}^{(i)}(N-1,k_i)} - Z \geq R(N) \geq N\frac{G_{opt}^{(i)}(N,k_i)}{G_{opt}^{(i)}(N-1,k_i)} - Z \tag{6-17}$$

$$\frac{N}{Z+R_{LB}^{(i)}(N)} \geq X(N) \geq \frac{N}{Z+R_{UB}^{(i)}(N)} \tag{6-18}$$

and, for each fixed-rate service center k,

$$X_{UB}^{(i)}(N)L_k \geq U_k(N) \geq X_{LB}^{(i)}(N)L_k \tag{6-19}$$

Proof:

Relations (6-18) and (6-19) follow from relation (6-17) by Little's equation. Since

$$R(N) = N\frac{G(N,k_i)}{G(N-1,k_i)} - Z$$

(where $G(N,k_0)$ is taken as the exact normalization constant for the queueing network formed by the service centers not included in $\{k_1, \cdots, k_i\}$), relation (6-17) and the theorem will be shown if it can be shown that

$$\frac{G_{opt}^{(i)}(N-1,k_i)}{G_{opt}^{(i)}(N,k_i)} \geq \frac{G(N-1,k_i)}{G(N,k_i)} \geq \frac{G_{pess}^{(i)}(N-1,k_i)}{G_{pess}^{(i)}(N,k_i)}$$

From lemma 6.2, this relation holds if $G_{opt}^{(0)}(0,k_0) = G_{pess}^{(0)}(0,k_0) = 1$, and if, for

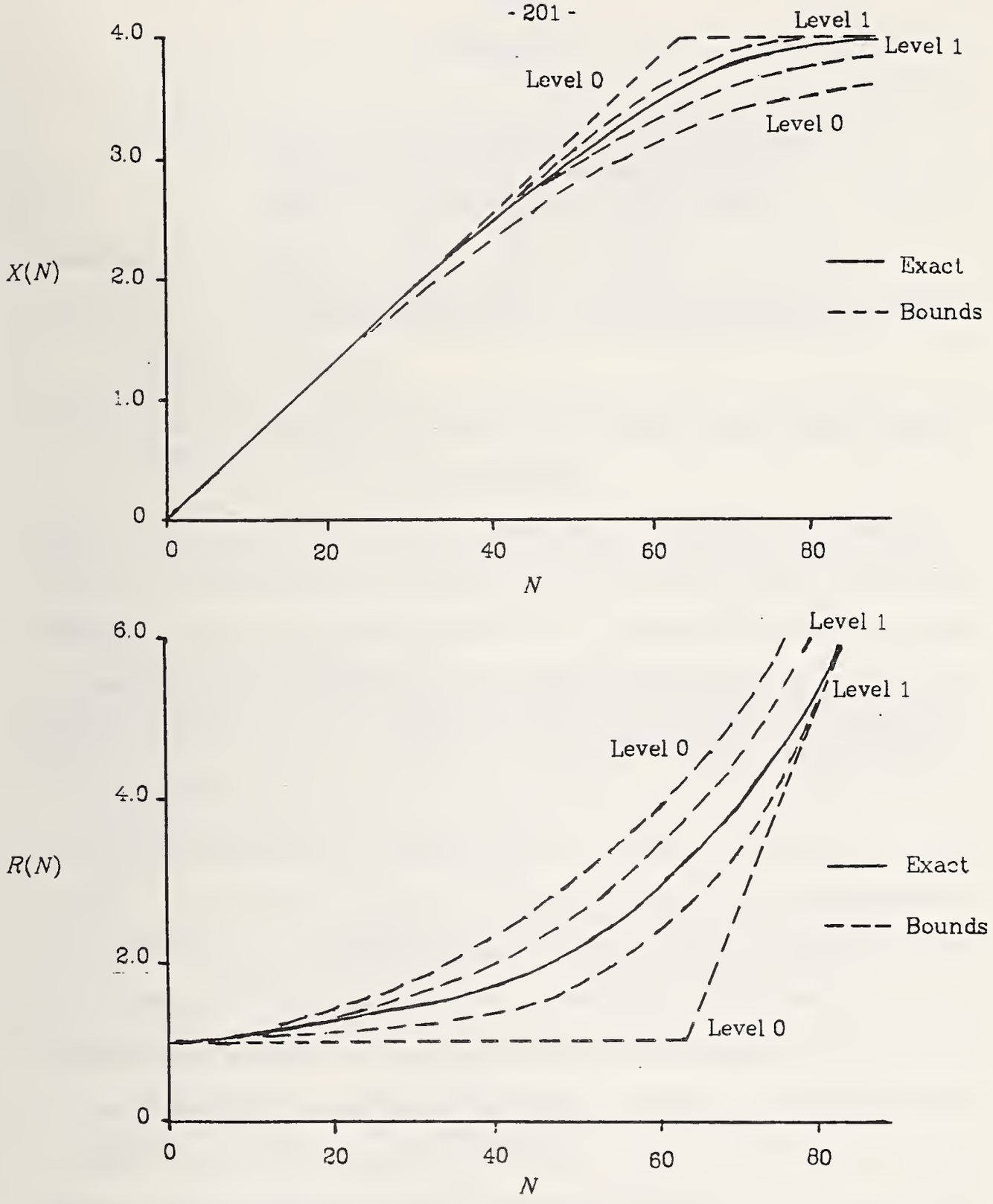
$1 \leq n \leq N$,

$$\frac{G_{opt}^{(0)}(n-1,k_0)}{G_{opt}^{(0)}(n,k_0)} \geq \frac{G(n-1,k_0)}{G(n,k_0)} \geq \frac{G_{pess}^{(0)}(n-1,k_0)}{G_{pess}^{(0)}(n,k_0)}$$

These relations hold from the choice of initial normalization constant estimates and the fact that $\frac{G(n-1,k_0)}{G(n,k_0)} = X^{(-k_1)\cdots(-k_i)}(n)$, establishing the theorem. ∎

Table 6.3 gives level 0, 1, 2 and 3 bounds yielded by the example bounding algorithm, for the queueing network of table 3.1 and a population of 50 customers. The level 0 bounds shown are identical to the level 0 PBH bounds, since these were used to compute the initial normalization constant estimates. The set $\{k_1, \cdots, k_i\}$ was chosen so that it included the i non-delay service centers with the largest loadings. (This choice is likely to be a good one in computer system modelling applications, since a computer system model usually includes a few heavily loaded service centers that are much more important in determining performance than the remaining service centers.) In figure 6.1, bounds on system throughput and mean system residence time are shown over a range of workload intensities; these bounds were computed in the same manner as those in table 6.3.

The accuracy of the example bounding algorithm based on service center recursion Convolution can be quite good. For a given accuracy of the initial normalization constant estimates, the accuracy is best when queueing network performance is almost completely determined by the characteristics of the service centers included in the set $\{k_1, \cdots, k_i\}$. In figure 6.1, this is the case as the queueing network nears saturation. For queueing networks with more highly skewed loadings, however, this will occur at more moderate workload intensities. In contrast to the PBH algorithm, the algorithm therefore performs well when loadings are highly skewed. The algorithm also has a



Bounds Yielded by an Algorithm Based on Service Center Iteration Convolution

Figure 6.1

Bound Level	Bounds on $R(50)$		Bounds on $X(50)$	
	Upper	Lower	Upper	Lower
0	3.10	1.00	3.13	2.76
1	2.49	1.61	3.01	2.86
2	2.31	1.85	2.97	2.89
3	2.27	1.99	2.94	2.89

Bounds Yielded by an Algorithm Based on Service Center Recursion Convolution

Table 6.3

significant advantage in that it can accurately treat load dependence. (The single closed class PBH algorithm can treat load dependence, except in special cases, only by specializing the multiple closed class PBH algorithm equations to single class queueing networks.)

The computational cost of the example bounding algorithm is, however, likely to be significantly greater than that of the PBH algorithm. For each load-dependent or delay service center that is included in the set $\{k_1, \cdots, k_i\}$, a term of N^2 must be added to the time complexity (unless such a service center is last in the ordering, in which case the term is only $4N$), while each fixed-rate service center contributes a term of $2N$. The total time complexity of the algorithm is therefore of the form $(c_1+2i_1+4i_2)N + c_2(K-i) + i_3N^2$, where $i_1+i_2+i_3$ equals the bound level i, and c_1 and c_2 depend on the level of the PBH algorithm bounds that are used in forming the initial normalization constant estimates. The space complexity of the algorithm is dominated by a term of N.

APPENDIX 6.A

This appendix states and proves lemmas of relevance to chapter six.

Lemma 6.1:

Suppose that two vectors of positive normalization constant values, $(G^{v(0)}(N,1), \cdots, G^{v(0)}(N,K))$ and $(G^{v'(0)}(N,1), \cdots, G^{v'(0)}(N,K))$, are such that

$$\frac{G^{v(0)}(N,k)}{G^{v(0)}(N,j)} \leq \frac{G^{v'(0)}(N,k)}{G^{v'(0)}(N,j)}$$

for all k, j such that $1 \leq k \leq j \leq K$. Then for all $i \geq 0$,

$$\frac{G^{v(i)}(N+i,k)}{G^{v(i)}(N+i,j)} \leq \frac{G^{v'(i)}(N+i,k)}{G^{v'(i)}(N+i,j)} \qquad (6.1\text{-}1)$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by i applications of the customer recursion Convolution equations starting from the values superscripted by $v(0)$ and $v'(0)$, respectively.

Proof:

The proof is by induction on i.

By assumption, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ $(i \geq 0)$. For $1 \leq k \leq j \leq K$, from the customer recursion Convolution equations relation (6.1-1) reduces to

$$\frac{\sum_{l=1}^{k} L_l G^{v(i)}(N+i,l)}{\sum_{l=1}^{j} L_l G^{v(i)}(N+i,l)} \leq \frac{\sum_{l=1}^{k} L_l G^{v'(i)}(N+i,l)}{\sum_{l=1}^{j} L_l G^{v'(i)}(N+i,l)}$$

$$\Longleftrightarrow \quad \frac{1}{1+\sum_{l_1=k+1}^{j} \frac{L_{l_1} G^{v(i)}(N+i,l_1)}{\sum_{l_2=1}^{k} L_{l_2} G^{v(i)}(N+i,l_2)}} \leq \frac{1}{1+\sum_{l_1=k+1}^{j} \frac{L_{l_1} G^{v'(i)}(N+i,l_1)}{\sum_{l_2=1}^{k} L_{l_2} G^{v'(i)}(N+i,l_2)}}$$

This last relation holds if, for each l_1 such that $k < l_1 \leq j$,

$$\frac{L_{l_1} G^{v(i)}(N+i,l_1)}{\sum_{l_2=1}^{k} L_{l_2} G^{v(i)}(N+i,l_2)} \geq \frac{L_{l_1} G^{v'(i)}(N+i,l_1)}{\sum_{l_2=1}^{k} L_{l_2} G^{v'(i)}(N+i,l_2)}$$

which in turn holds if, for each l_2 such that $1 \leq l_2 \leq k < l_1 \leq j$,

$$\frac{G^{v(i)}(N+i,l_2)}{G^{v(i)}(N+i,l_1)} \leq \frac{G^{v'(i)}(N+i,l_2)}{G^{v'(i)}(N+i,l_1)}$$

This last relation holds from the inductive hypothesis, establishing relation (6.1-1) and the lemma for $i+1$. By induction, the lemma is established for $i \geq 0$. ∎

Lemma 6.2:

Suppose that positive normalization constant values $G^{v(0)}(n,k)$ and $G^{v'(0)}(n,k)$ $(1 \le n \le N)$ are such that

$$G^{v(0)}(0,k) = G^{v'(0)}(0,k) = 1$$

and, for $1 \le n \le N$,

$$\frac{G^{v(0)}(n-1,k)}{G^{v(0)}(n,k)} \ge \frac{G^{v'(0)}(n-1,k)}{G^{v'(0)}(n,k)}$$

If $\frac{L_j(m)}{m} \ge \frac{L_j(m+1)}{m+1}$ for $1 \le m \le N-1$ and each load-dependent service center j, then for $i \ge 0$,

$$G^{v(i)}(0,k+i) = G^{v'(i)}(0,k+i) = 1 \qquad (6.2\text{-}1)$$

and, for $1 \le n \le N$,

$$\frac{G^{v(i)}(n-1,k+i)}{G^{v(i)}(n,k+i)} \ge \frac{G^{v'(i)}(n-1,k+i)}{G^{v'(i)}(n,k+i)} \qquad (6.2\text{-}2)$$

where the superscripts $v(i)$ and $v'(i)$ denote values yielded by i applications of the service center recursion Convolution equations starting from the values superscripted by $v(0)$ and $v'(0)$, respectively.

Proof:

The proof is by induction on i.

By assumption, the lemma holds for the induction basis $i=0$. Assume that the lemma holds for i, and consider the lemma for $i+1$ $(i \ge 0)$. Define $P_{k+i+1}(m)$ by

$$P_{k+i+1}(m) = L^m_{k+i+1}$$

if $k+i+1$ indexes a fixed-rate service center,

$$P_{k+i+1}(m) = \frac{\prod_{l=1}^{m} L_{k+i+1}(l)}{m!}$$

if $k+i+1$ indexes a load-dependent service center, and by

$$P_{k+i+1}(m) = \frac{Z^m}{m!}$$

if $k+i+1$ indexes an aggregate delay service center. From equations (6-1), (6-12) and (6-13), for $0 \le n \le N$

$$G^{v(i+1)}(n,k+i+1) = \sum_{m=0}^{n} G^{v(i)}(n-m,k+i)P_{k+i+1}(m)$$

and

$$G^{v'(i+1)}(n,k+i+1) = \sum_{m=0}^{n} G^{v'(i)}(n-m,k+i)P_{k+i+1}(m)$$

Relation (6.2-1) for $i+1$ follows directly from the inductive hypothesis. Relation (6.2-2) for $i+1$ reduces to, for $1 \le n \le N$,

$$\frac{\sum_{m=0}^{n-1} G^{v(i)}(n-1-m,k+i)P_{k+i+1}(m)}{\sum_{m=0}^{n} G^{v(i)}(n-m,k+i)P_{k+i+1}(m)} \ge \frac{\sum_{m=0}^{n-1} G^{v'(i)}(n-1-m,k+i)P_{k+i+1}(m)}{\sum_{m=0}^{n} G^{v'(i)}(n-m,k+i)P_{k+i+1}(m)}$$

$$<==> \left(\sum_{m=0}^{n-1} G^{v(i)}(n-1-m,k+i)P_{k+i+1}(m)\right)\left(\sum_{m=0}^{n} G^{v'(i)}(n-m,k+i)P_{k+i+1}(m)\right) \geq$$

$$\left(\sum_{m=0}^{n} G^{v(i)}(n-m,k+i)P_{k+i+1}(m)\right)\left(\sum_{m=0}^{n-1} G^{v'(i)}(n-1-m,k+i)P_{k+i+1}(m)\right) \qquad (6.2\text{-}3)$$

Since relation (6.2-2) is transitive, relation (6.2-3) need only be shown for normalization constants $G^{v(i)}(n,k+i)$ such that

$$\frac{G^{v(i)}(n-1,k+i)}{G^{v(i)}(n,k+i)} = \frac{G^{v'(i)}(n-1,k+i)}{G^{v'(i)}(n,k+i)}$$

for all $n \neq n^*$, for some n^* such that $1 \leq n^* \leq N$. In conjunction with relation (6.2-1), this yields

$$G^{v(i)}(n,k+i) = G^{v'(i)}(n,k+1) \qquad 0 \leq n < n^*$$

and

$$G^{v(i)}(n,k+i) = cG^{v'(i)}(n,k+1) \qquad n^* \leq n \leq N$$

for some constant $c \leq 1$. If $c=1$, relation (6.2-3) is trivially true, so consider $c<1$. Substitution in relation (6.2-3) yields, after canceling terms and dividing through by $1-c$,

$$\left(\sum_{m=n-n^*}^{n-1} G^{v'(i)}(n-1-m,k+i)P_{k+i+1}(m)\right)\left(\sum_{m=0}^{n-n^*} G^{v'(i)}(n-m,k+i)P_{k+i+1}(m)\right) \geq$$

$$\left(\sum_{m=n-n^*+1}^{n} G^{v'(i)}(n-m,k+i)P_{k+i+1}(m)\right)\left(\sum_{m=0}^{n-n^*-1} G^{v'(i)}(n-1-m,k+i)P_{k+i+1}(m)\right)$$

(where empty sums are defined as 0). This relation holds if, for $0 \leq m_1 \leq n-n^*-1$ and $n-n^* \leq m_2 \leq n-1$,

$$P_{k+i+1}(m_2)P_{k+i+1}(m_1+1) \geq P_{k+i+1}(m_2+1)P_{k+i+1}(m_1)$$

$$<==> \frac{P_{k+i+1}(m_2)}{P_{k+i+1}(m_2+1)} \geq \frac{P_{k+i+1}(m_1)}{P_{k+i+1}(m_1+1)}$$

Noting that $m_2 > m_1$, this last relation holds from the definition of $P_{k+i+1}(m)$ and the assumption regarding the loadings of a load-dependent service center. This establishes relation (6.2-2) and the lemma for $i+1$. By induction, the lemma is established for $i \geq 0$. ∎

CHAPTER 7

A COMPARISON AMONG BOUNDING ALGORITHMS

An important context in which to evaluate the relative merits of bounding algorithms is that of large queueing networks with multiple customer classes. It is infeasible to obtain the exact solutions of these networks, and, as noted in chapter two, approximation techniques may induce unacceptable errors.

The PBH and LPBH bounding algorithms are both applicable to queueing networks with multiple customer classes, as is PANACEA, one of the bounding algorithms that was surveyed in chapter two. In this chapter, these three bounding algorithms are compared with respect to their applicability (section one), their implementation effort (section two), and their accuracy and computational cost (section three). Section four uses the results of these comparisons to suggest which bounding algorithm is best suited for various accuracy requirements, cost constraints, and queueing network types.

7.1. Applicability

PANACEA and the PBH and LPBH algorithms are only applicable to separable queueing networks. In addition, none of these algorithms has been made sufficiently general to handle all types of these networks.

The queueing networks treated by the PBH and LPBH algorithms may contain fixed-rate service centers, delay centers, and load-dependent centers at which the service rate depends only on the total number of customers present at the center. Other types of separable queueing networks allow more general

forms of load dependence, and currently cannot be treated.

PANACEA treats queueing networks with fixed-rate and delay service centers. Each customer class must visit a delay service center; this requirement is part of a complex and not easily defined restriction that PANACEA places on the queueing network parameter values. One statement of this restriction is that $\frac{L_{ks}}{Z_s}$ must be on the order of $\frac{1}{N_s}$ for all fixed-rate service centers k and classes s [McKenna & Mitra 81, McKenna, Mitra & Ramakrishnan 81]. However, this statement is somewhat ambiguous; in its most strict interpretation it would require, for example, all of the fixed-rate service center loadings of each class to be of the same order of magnitude. Another, not equivalent, statement of PANACEA's restrictions is that the utilization of each service center must be less than 85 percent [Ramakrishnan & Mitra 82]. However, experimental results (section 7.3) suggest that the range of utilizations treated by the algorithm is strongly dependent on the delay service center loadings, increasing as Z_s for each class s increases. For parameter values that the algorithm could not properly handle, version 2.0 of PANACEA, which was used to obtain experimental results, either gave no bounds, or gave bounds that were less accurate than even simple asymptotic bounds.

It is, of course, possible that PANACEA, the PBH algorithm, or the LPBH algorithm can be extended to treat more general types of queueing networks. In particular, since the PBH and LPBH algorithms are based on the MVA solution algorithm, it seems quite possible that they could be extended, as has been previously done with MVA [Sauer 83], to treat general separable queueing networks.

7.2. Implementation Effort

The complexity of the theory underlying PANACEA makes it doubtful that the algorithms used could be independently implemented without significant

difficulties. Version 1.1 of PANACEA consists of approximately 6000 lines of C-code [Ramakrishnan & Mitra 82].

In contrast, the PBH and LPBH algorithms are based on the widely known structure of the MVA solution algorithm. The implementations of the PBH and LPBH algorithms that were used to provide numerical results consist of approximately 200 and 300 lines of C-code, respectively. (However, these implementations provided a user interface much simpler than that provided by PANACEA.)

7.3. Accuracy and Computational Cost

The computational costs of the PBH and LPBH algorithms were given in tables 4.3 and 5.13, respectively. Both algorithms were applied to the same test queueing networks, with results as shown in tables 4.4 to 4.9, and 5.15 to 5.19. In the accuracy tests, the level i LPBH bounds were of approximately the same accuracy as the level $i+2$ PBH bounds. The computational cost results show that the former bounds are more costly in terms of time, but less costly in terms of space. This cost relationship was also observed empirically.

Like the PBH and LPBH algorithms, PANACEA offers a tradeoff between computational effort and accuracy. In PANACEA, this tradeoff is provided by the number of terms chosen in the asymptotic expansions of the required normalization constants. Unfortunately, in version 2.0 of PANACEA the number of terms is chosen by the software rather by the user. In addition, there is an upper limit of 4 on this quantity [Ramakrishnan & Mitra 82].

It is difficult to determine the computational complexity of the algorithm used by PANACEA. Given the upper limit of 4 on the number of terms t chosen in the asymptotic expansions, the time complexity is on the order of $K|C|^t$, while the storage requirement is approximately $20K$ kilobytes [Ramakrishnan & Mitra 82]. However, it is not clear how the time and space complexities would

vary if t was increased beyond 4. Empirically, the choice of t=4 (which PANACEA consistently made in the accuracy tests) corresponded, in terms of computation time, to a PBH bound level of at least 3.

The accuracy of PANACEA was compared to that of the PBH algorithm by applying both algorithms to the queueing network of table 7.1, and to the network resulting from doubling the delay service center loadings. (PANACEA is not applicable to the queueing networks used in the previous accuracy tests of the PBH and LPBH algorithms, since these networks have customer classes that do not visit delay service centers.) The results of these tests are shown in tables 7.2 to 7.5. Only the primary performance measures of system throughput and mean system residence time are considered. The values given for PANACEA were not directly obtainable, and had to be derived by applying Little's equation to the bounds provided by PANACEA on service center utilizations.

	Fixed-rate Service Center Loadings								Delay Service Center Loadings
Class	1	2	3	4	5	6	7	8	
A	.25	.20	.17	.16	.08	.06	.05	.03	10
B	.08	.16	.17	.25	.03	.20	.05	.06	15
C	.03	.05	.25	.08	.17	.16	.06	.20	20
D	.20	.05	.08	.03	.25	.06	.16	.17	30

An Example Queueing Network

Table 7.1

Like the PBH algorithm, PANACEA's accuracy became poorer as the queueing network congestion increased. Unlike the PBH algorithm, PANACEA's accuracy improved when the delay service center loadings were doubled, for a given level of queueing network congestion. For example, note that all throughputs

System Throughput Bounds								
Populations	Class *A*		Class *B*		Class *C*		Class *D*	
	Upper	Lower	Upper	Lower	Upper	Lower	Upper	Lower
10 10 20 20	.87	.81	.61	.58	.93	.89	.64	.62
10 10 10 40	.87	.79	.61	.57	.46	.44	1.27	1.22
10 10 10 60	.86	.71	.60	.53	.46	.41	1.89	1.75

Mean System Residence Time Bounds								
Populations	Class *A*		Class *B*		Class *C*		Class *D*	
	Upper	Lower	Upper	Lower	Upper	Lower	Upper	Lower
10 10 20 20	2.37	1.51	2.39	1.50	2.43	1.52	2.41	1.48
10 10 10 40	2.65	1.54	2.63	1.48	2.68	1.52	2.74	1.56
10 10 10 60	4.06	1.63	4.01	1.54	4.13	1.62	4.25	1.71

Level 3 PBH Bounds for an Example Queueing Network

Table 7.2

System Throughput Bounds								
Populations	Class *A*		Class *B*		Class *C*		Class *D*	
	Upper	Lower	Upper	Lower	Upper	Lower	Upper	Lower
10 10 20 20	.47	.47	.32	.32	.48	.48	.33	.33
20 20 40 40	.93	.87	.63	.61	.96	.93	.65	.64
20 20 40 80	.92	.80	.63	.57	.96	.88	1.30	1.23

Mean System Residence Time Bounds								
Populations	Class *A*		Class *B*		Class *C*		Class *D*	
	Upper	Lower	Upper	Lower	Upper	Lower	Upper	Lower
10 10 20 20	1.33	1.25	1.33	1.24	1.34	1.25	1.32	1.23
20 20 40 40	2.94	1.56	2.95	1.54	2.99	1.57	2.96	1.52
20 20 40 80	5.10	1.65	5.10	1.60	5.21	1.68	5.24	1.68

Level 3 PBH Bounds for an Example Queueing Network
(Doubled Delay Service Center Loadings)

Table 7.3

System Throughput Bounds								
Populations	Class *A*		Class *B*		Class *C*		Class *D*	
	Upper	Lower	Upper	Lower	Upper	Lower	Upper	Lower
10 10 20 20	.91	.79	.62	.58	1.00	.86	.66	.64
10 10 10 40	.92	.76	.62	.57	.47	.45	1.33	1.20
10 10 10 60	1.00	-	.67	.22	.50	.23	2.00	1.16

Mean System Residence Time Bounds								
Populations	Class *A*		Class *B*		Class *C*		Class *D*	
	Upper	Lower	Upper	Lower	Upper	Lower	Upper	Lower
10 10 20 20	2.69	1.02	2.36	1.19	3.24	0.00	1.08	0.44
10 10 10 40	3.12	.87	2.41	1.14	2.37	1.26	3.29	0.00
10 10 10 60	-	0.00	30.05	0.00	24.39	0.00	21.72	0.00

PANACEA Bounds for an Example Queueing Network

Table 7.4

System Throughput Bounds								
Populations	Class *A*		Class *B*		Class *C*		Class *D*	
	Upper	Lower	Upper	Lower	Upper	Lower	Upper	Lower
10 10 20 20	.47	.47	.32	.32	.49	.49	.33	.33
20 20 40 40	.92	.92	.63	.63	.97	.97	.65	.65
20 20 40 80	.92	.90	.63	.62	.97	.96	1.33	1.27

Mean System Residence Time Bounds								
Populations	Class *A*		Class *B*		Class *C*		Class *D*	
	Upper	Lower	Upper	Lower	Upper	Lower	Upper	Lower
10 10 20 20	1.26	1.26	1.26	1.25	1.21	1.21	1.19	1.18
20 20 40 40	1.82	1.68	1.77	1.67	1.38	1.28	1.30	1.26
20 20 40 80	2.34	1.80	2.12	1.76	1.49	1.15	3.12	0.00

PANACEA Bounds for an Example Queueing Network
(Doubled Delay Service Center Loadings)

Table 7.5

and therefore utilizations for a population vector of (20 20 40 80) and with doubled delay service center loadings are greater than those for a population vector of (10 10 10 40) with the original delay service center loadings, yet PANACEA's accuracy is much better in the former case.

It is difficult to make more precise statements about the comparative accuracy of the two algorithms because of a problem with bound reliability in version 2.0 of PANACEA. There are several instances in tables 7.2 to 7.5 where the PANACEA bounds do not overlap with the corresponding PBH bounds. To verify that is was indeed the PANACEA bounds that were in error, the queueing network solution package THEsolver [THEsolver 81] was used to obtain exact results (at considerable cost, requiring approximately an hour of CPU time on an IBM 3033) for the queueing network of table 7.1 and a population vector of (10 10 20 20). The exact class D system throughput and mean system residence time values were obtained as .63 and 1.59, respectively, both of which lie outside the bounds given by PANACEA.

7.4. Recommendations

It appears that the PBH and LPBH algorithms are to be recommended over PANACEA as general purpose modelling tools, principally due to their treatment of a much larger class of queueing networks. PANACEA can be recommended only when the modeller's accuracy requirements justify PANACEA's implementation, time and storage costs, and only for queueing networks in which each class has a large delay service center loading, and in which there are no load-dependent service centers. (This recommendation is premised on the assumption that the problems with bound reliability noted in the previous section can be overcome.)

The choice between the PBH and LPBH algorithms is essentially one of time versus space. The level i LPBH bounds are empirically of similar accuracy as

the level $i+2$ PBH bounds, and are more costly in terms of computation time, but less costly in terms of space. In general, the PBH algorithm is recommended when time is relatively expensive with respect to space, while the LPBH algorithm is recommended when space is relatively expensive with respect to time. An additional factor which may affect the choice between the algorithms is the existence of a symbolic formulation for the PBH algorithm, but not for the LPBH algorithm. A symbolic formulation is particularly useful if queueing network performance measures are required for many different population levels.

CHAPTER 8

CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

This chapter summarizes the thesis contents (section one) and suggests directions for future research (section two).

8.1. Summary of Thesis Work

The cost of using an exact solution algorithm to analyze separable queueing networks may be prohibitive, due to either the number or the size of the networks that must be solved. Approximation techniques can be used to reduce this cost, but they introduce additional errors. The modeller's confidence in the model validation process, and in the performance predictions obtained from a model, can be increased by providing bounds on these errors.

This thesis has proposed a number of algorithms that first compute bounds on the queueing network solution, and then apply a simple heuristic to obtain point estimates. Error bounds are thus easily obtainable. Each of the proposed bounding algorithms is based on the recursive structure of one of the exact solution algorithms, in which a queueing network solution is recursively computed from the solutions of simpler networks. By utilizing recursion basis queueing networks that are much more complex than the solution algorithm recursion basis networks, the computational cost is significantly reduced. Appropriate estimates of the performance measures of these more complex networks, and in some cases appropriate modifications of the solution algorithm equations, guarantee that the computation yields performance measure

bounds. One advantage of this structure is that the choice of the complexity of the recursion basis networks provides a tradeoff between computational cost and accuracy.

The bounding algorithms that have been proposed in this thesis differ with respect to the type of queueing network treated, the solution algorithm used to provide the basic computational structure, and the precise manner in which the solution algorithm structure is utilized. Each of these aspects will be treated here in turn.

Bounding algorithms for queueing networks with only a single closed class and optionally some number of open classes were proposed in chapters three, five and six. These algorithms are primarily of theoretical interest, as it is, in practice, always feasible to solve a single closed class queueing network exactly. (These algorithms are of practical benefit only when the number of queueing networks that are to be analyzed is large.) In particular, the development of the single closed class bounding algorithms provided insight into the design and analysis of multiple closed class bounding algorithms, as well as theoretical results of independent importance such as the convexity of mean system residence time.

Bounding algorithms for queueing networks with multiple closed classes and optionally some number of open classes were proposed in chapters four and five. These algorithms were developed as extensions of the single closed class algorithms; although these extensions required substantial modifications, the basic ideas remained the same.

The bounding algorithms proposed in chapters three through five were based on the Mean Value Analysis solution algorithm (both the symbolic and the non-symbolic formulation), while those proposed in chapter six were based on the Convolution solution algorithm. Mean Value Analysis was found to be more

suitable than Convolution as a bounding algorithm basis, which is consistent with the former algorithm's popularity as a basis for approximation techniques.

The bounding algorithms based on Mean Value Analysis can be further differentiated by whether or not they employ looping (chapter five) in their computation of initial performance measure estimates. By using approximations that give the performance measures of the populations with one fewer customer (one population for each class) in terms of those for the original population, an iteration, or looping, can be performed between the performance measures of adjacent population levels. Applied to multiple closed class queueing networks, looping saves space at the expense of time, for a given degree of accuracy.

The bounding algorithms proposed in this thesis are quite different in structure from PANACEA, the major alternative bounding algorithm that is also applicable to queueing networks with multiple closed customer classes. In chapter seven, comparisons were made among PANACEA and the multiple closed class bounding algorithms proposed in this thesis. The latter bounding algorithms were found to have a number of advantages with respect to PANACEA, most notably their treatment of a much larger class of queueing networks.

This thesis has also proposed, in chapter four, an error-bounded approximation algorithm applicable to multiple closed class queueing networks. An error-bounded approximation algorithm differs from a bounding algorithm in that a major portion of its computational effort is expended on point estimates. Bounds on the point estimate errors, if desired, are optionally provided after the point estimates have been obtained. In the algorithm proposed in this thesis, error bounds are provided by a modification of the multiple closed class PBH algorithm.

8.2. Directions for Future Research

It remains to be seen whether or not bounding or error-bounded approximation algorithms will become popular as practical tools. The issue is primarily one of psychology: does the increased confidence afforded by error bounds justify the additional cost required to obtain them?

The answer to this question is, of course, dependent on the quality of the error bounds. For single closed class queueing networks, the algorithms proposed in this thesis can be used to provide pessimistic bounds on primary performance measures, for example, at the same or at a lower cost than approximations of similar accuracy that lack error bounds. For multiple closed class queueing networks, however, point estimates can be obtained with less computational effort than PBH or LPBH bounds of similar accuracy. One important research question is whether the multiple closed class bounding algorithms proposed in this thesis (in particular, the looped algorithm) can be improved to diminish or remove this cost penalty.

The usefulness of error bounds also depends on the possible magnitude of the errors induced by the popular approximation algorithms. Although it is known that the aMVA/PE algorithm can induce large errors, it is not known if large errors are possible in more computationally costly algorithms such as Linearizer. However, one advantage of the PBH and LPBH algorithms is their provision of tradeoffs between accuracy and computational cost. In particular, they can provide bounds at a much lower computational cost than Linearizer can provide point estimates (although these bounds may be much less accurate).

The practical utility of bounding algorithms can also be improved by increasing their applicability. Many features important to computer system performance are best modelled in certain types of non-separable queueing

networks, for which a number of useful approximation techniques have recently been developed. Previous research [Sevcik 77] suggests the feasibility of developing algorithms that empirically and/or intuitively provide bounds for these networks, but there has been little research that offers useful proof techniques. Much work in this area remains to be done.

APPENDIX A

NOTATION SUMMARY

Where applicable, quantities are indexed by the queueing network workload intensity for which they apply. In the context of single customer class queueing networks, class subscripts are omitted.

$a_s(\vec{N},\vec{\lambda})$	Optimistic asymptotic bound on class s mean system residence time.
A_k	For a fixed-rate service center k, $\lvert\{j : L_{js} \geq L_{ks}$ for all $s \in C$, j fixed-rate service center$\}\rvert$.
$b_s(\vec{N},\vec{\lambda})$	An extension to queueing networks with delay service centers of a Balanced Job Bound algorithm pessimistic bound on class s mean system residence time.
c	Index of the closed class (single closed class context only).
c_k	The capacity of a load-dependent service center k (the total service rate is constant when c_k or more customers are present).
C	Set indexing the closed classes.
$\vec{e}_s$	A queueing network population vector consisting of only one class s customer.
$E_k(\vec{N})$	Expansion ratio for service center k. For an arbitrary closed class t such that $L_{kt} \neq 0$, this quantity is defined as $\frac{R_{kt}(\vec{N}+\vec{e}_t)}{L_{kt}}$.
$G(\vec{N},\vec{\lambda},k)$	Normalization constant for the queueing network consisting of service center k and those service centers preceding center k in some indexing.
K	Number of queueing network service centers, excluding delay service centers.
L_{ks}	Loading of a class s customer at a fixed-rate service center k.
$L_{ks}(m)$	Loading of a class s customer at a load-dependent service center k when there are m customers present.

L_{sr}^{h} — The average loading of class s customers when encountering customers from the class r heap.

L_{s}^{h} — The average loading of class s customers when encountering customers from a heap (class unspecified).

$L^{(p)}$ — $\sum_{k=1}^{K} L_k^p$ (single closed class, no load-dependent service center context only).

$L^{[s_1,\cdots,s_p]}$ — $\sum_{k=1}^{K}\prod_{g=1}^{p} L_{ks_g}$ (multiple closed classes, no load-dependent service center context only).

$n_{ks}(\vec{N},\vec{\lambda})$ — Mean queue length of class s customers at service center k.

$n_{s}^{h}(\vec{N},\vec{\lambda})$ — Class s queue length (heap) that is to be allocated to a group of queueing network service centers, and that has not been accounted for by some choice of approximate mean queue lengths.

N_s — Number of customers (workload intensity) of a closed class s.

N — Total number of customers in the closed classes.

$\vec{N}$ — A vector giving the number of customers of each closed class.

O — Set indexing the open classes.

$p_k(m \mid \vec{N},\vec{\lambda})$ — Probability of m customers enqueued at service center k.

$prec_r$ — $\sum_{l} m_l$ for l indexing the classes that precede class r in some arbitrary ordering of the elements of C (m_l values fixed by context).

$R_{ks}(\vec{N},\vec{\lambda})$ — Mean total residence time of a class s customer at service center k.

$R_s(\vec{N},\vec{\lambda})$ — Mean system residence time (excluding delay service center mean residence time) of a class s customer.

$U_{ks}(\vec{N},\vec{\lambda})$ Utilization of service center k by class s customers.

$V_k(n,p)$ $L_k^p E_k(n-1)$ (single closed class, no load-dependent service center context only).

$V(n,p)$ $\sum_{k=1}^{K} L_k^p E_k(n-1)$ (single closed class, no load-dependent service center context only).

$W_k(n,p)$ $L_k^p E_k(n-p)$ (single closed class, no load-dependent service center context only).

$W(n,p)$ $\sum_{k=1}^{K} L_k^p E_k(n-p)$ (single closed class, no load-dependent service center context only).

$W_k(\vec{n},[s_1,\cdots,s_p])$ $(\prod_{g=1}^{p} L_{ks_g}) E_k(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g})$ (multiple closed classes, no load-dependent service center context only).

$W(\vec{n},[s_1,\cdots,s_p])$ $\sum_{k=1}^{K}(\prod_{g=1}^{p} L_{ks_g}) E_k(\vec{n}-\sum_{g=1}^{p}\vec{e}_{s_g})$ (multiple closed classes, no load-dependent service center context only).

$X_s(\vec{N},\vec{\lambda})$ System throughput of class s customers.

Z_s Class s loading at an aggregate delay service center.

δ_{rs} Equals 1 if $r=s$; 0 otherwise.

$\Delta(N)$ The change in the mean number of customers at the fixed-rate service centers when going from population level $N-1$ to N (single closed class context only).

λ_s Arrival rate of customers (workload intensity) of an open class s.

$\vec{\lambda}$ A vector giving the arrival rate of each open class.

Special Superscripts

$v(i)$ Performance measures yielded by i computational steps from initial performance measure values superscripted by $v(0)$ (v may be used as a shortened form of $v(0)$).

(i)	Performance measures yielded by i computational steps (initial performance measure values either clear from context, or not specified).
$+k$	Performance measures of a queueing network with an additional service center with loadings identical to those of service center k.
$-k$	Performance measures of a queueing network with service center k removed.

Special Subscripts

appr	Quantities produced by an error-bounded approximation algorithm.
b	The index of a bottleneck service center (single closed class, no load-dependent service center context only).
LB	Lower bound on a quantity.
opt	Quantities produced during a computation of optimistic performance measure bounds.
pess	Quantities produced during a computation of pessimistic performance measure bounds.
UB	Upper bound on a quantity.

REFERENCES

[Agrawal 83]
Agrawal, S.C. Metamodeling: A study of approximations in queueing models. Ph.D. Thesis, Purdue University, West Lafayette, IN., Aug. 1983.

[Au 81]
Au, M. Some properties of queueing network models. M.Sc. Thesis, University of Toronto, Toronto, Ont., 1981.

[Barbour 76]
Barbour, A.D. Networks of queues and the method of stages. *Adv. Appl. Prob.* 8 (1976), 584-591.

[Bard 78]
Bard, Y. An analytic model of the VM/370 system. *IBM J. Res. and Dev.* 22, 5 (Sept. 1978), 498-508.

[Bard 79]
Bard, Y. Some extensions to multiclass queueing network analysis, in: M. Arato, A. Butrimenko and E. Gelenbe (eds.), *Performance of Computer Systems*. North Holland, Amsterdam, 1979.

[Bard 80]
Bard, Y. A model of shared DASD and multipathing. *Comm. ACM* 23, 10 (Oct. 1980), 564-572.

[Bard 81]
Bard, Y. A simple approach to system modeling. *Perf. Eval.* 1, 3 (Nov. 1981), 225-248.

[Baskett et al. 75]
Baskett, F., K.M. Chandy, R.R. Muntz, and F.G. Palacios. Open, closed and mixed networks of queues with different classes of customers. *J. ACM* 22, 2 (April 1975), 248-260.

[Bryant & Agre 81]
Bryant, R.M. and J.R. Agre. A queueing network approach to the module allocation problem in distributed systems. Proc. of the ACM/SIGMETRICS Conf. on Meas. and Modeling of Comp. Sys., Las Vegas, Sept. 1981, pp. 191-204.

[Buzen 73]
Buzen, J.P. Computational algorithms for closed queueing networks with exponential servers. *Comm. ACM* 16, 9 (Sept. 1973), 527-531.

[Buzen & Denning 80]
Buzen, J.P., and P.J. Denning. Operational treatment of queue distributions and Mean Value Analysis. *Computer Performance* 1, 1 (June 1980), 6-15.

[Chandy, Howard & Towsley 77]
Chandy, K.M., J.H. Howard Jr., and D.F. Towsley. Product form and local balance in queueing networks. *J. ACM* 24, 2 (April 1977), 250-263.

[Chandy & Martin 83]
Chandy, K.M., and A J. Martin. A characterization of product-form queuing networks. *J. ACM* 30, 2 (April 1983), 286-299.

[Chandy & Neuse 82]
Chandy, K.M., and D. Neuse. Linearizer: a heuristic algorithm for queueing network models of computing systems. *Comm. ACM* 25, 2 (Feb. 1982), 126-134.

[Chandy & Sauer 80]
Chandy, K.M., and C.H. Sauer. Computational algorithms for product form queueing networks. *Comm. ACM* 23, 10 (Oct. 1980), 573-583.

[Chow 83]
Chow, W.-M. Approximations for large scale closed queueing networks. *Perf. Eval.* 3, 1 (Feb. 1983), 1-12.

[Denning & Buzen 78]
Denning, P.J., and J.P. Buzen. The operational analysis of queueing network models. *ACM Computing Surveys* 10, 3 (Sept. 1978), 225-261.

[Dowdy et al. 83]
Dowdy, L.W., D.L. Eager, K.D. Gordon, and L.V. Saxton. Throughput concavity and response time convexity. Submitted for publication, 1983.

[Dowdy, Gordon & Saxton 83]
Dowdy, L.W., K.D. Gordon, and L.V. Saxton. A proof that throughput is concave. Tech. Rep. CS-83-02, Vanderbilt University, Nashville, TN., Feb. 1983.

[Dowdy, Perez-Davila & Stephens 83]
Dowdy, L.W., A. Perez-Davila, and L.E. Stephens. Performance bounds based upon throughput curve properties. Proc. of the Ninth International Symposium on Computer Performance Modelling, Measurement, and Evaluation, "Performance '83", College Park, May 1983, pp. 285-293.

[Eager 83]
Eager, D.L. A proof of throughput concavity and residence time convexity. Unpublished note, University of Toronto, Toronto, Ont., 1983.

[Eager & Sevcik 83a]
Eager, D.L., and K.C. Sevcik. Performance Bound Hierarchies for queueing networks. *ACM Trans. on Comp. Sys.* 1, 2 (May 1983), 99-115.

[Eager & Sevcik 83b]
Eager, D.L., and K.C. Sevcik. An analysis of an approximation algorithm for queueing networks. Submitted for publication, 1983.

[Graham 78]
Graham, G.S. (ed.) Special Issue: Queueing network models of computer system performance. *ACM Comp. Surv.* 10, 3 (Sept. 1978).

[Graham 81]
Graham, G.S. Asymptotic properties of mean device queue length. Unpublished report, University of Toronto, Toronto, Ont., 1981.

[Hoyme et al. 82]
Hoyme, K.P., S.C. Bruell, P.V. Afshari, and R.Y. Kain. A tree-structured MVA algorithm. Tech. Rep. 82-17, Computer Science Department, University of Minnesota, Minneapolis, Minnesota, Dec. 1982.

[Kelly 79]
Kelly, F.P. *Reversibility and Stochastic Networks*. John Wiley and Sons, New York, 1979.

[Kleinrock 76]
Kleinrock, L. *Queueing Systems: Volume* 2. John Wiley and Sons, New York, 1976.

[Kobayashi 79]
Kobayashi, H. A computational algorithm for queue distributions via the Polya theory of enumeration, in: M. Arato, A. Butrimenko and E. Gelenbe (eds.), *Performance of Computer Systems*. North Holland, Amsterdam, 1979.

[Kriz 82]
Kriz, J. Throughput bounds for closed queueing networks. Submitted for publication, 1982.

[Lam 82]
Lam, S.S. Dynamic scaling and growth behavior of queueing network normalization constants. *J. ACM* 29, 2 (April 1982), 492-513

[Lam & Lien 81]
Lam, S.S., and Y.L. Lien. An analysis of the tree convolution algorithm for queueing networks. Tech. Rep. TR-166, Department of Computer Sciences, University of Texas, Austin, TX., Jan. 1981.

[Lam & Lien 83]
Lam, S.S., and Y.L. Lien. A tree convolution algorithm for the solution of queueing networks. *Comm. ACM* 26, 3 (March 1983), 203-215.

[Lavenberg & Reiser 80]
Lavenberg, S.S., and M. Reiser. Stationary state probabilities at arrival instants for closed queueing networks with multiple types of customers. *J. Appl. Prob.* 17 (1980), 1048-1061.

[Lazowska et al. 84]
Lazowska, E.D., J. Zahorjan, G.S. Graham, and K.C. Sevcik. *Quantitative System Performance*. Prentice-Hall, Englewood Cliffs, N.J., 1984.

[Little 61]
Little, J.D.C. A proof of the queueing formula L = λW. *Oper. Res.* 9 (1961), 383-387.

[Marshall & Olkin 79]
Marshall, A.W. and I. Olkin. *Inequalities: Theory of Majorization and its Applications*. Academic Press, New York, 1979.

[McKenna & Mitra 81]
McKenna, J., and D. Mitra. Integral representations and asymptotic expansions for closed Markovian queueing networks: normal usage. Proc. of the Eighth International Symposium on Computer Performance Modelling, Measurement, and Evaluation, "Performance '81", Amsterdam, Nov. 1981, pp. 67-84.

[McKenna, Mitra & Ramakrishnan 81]
McKenna, J., D. Mitra and K.G. Ramakrishnan. A class of closed Markovian queueing networks: integral representations, asymptotic expansions, and generalizations. *Bell System Tech. J.* 60, 5 (May-June 1981), 559-641.

[Muntz & Wong 74]
Muntz, R.R., and J. Wong. Asymptotic properties of closed queueing network models. 8th. Annual Princeton Conf. on Inf. Sci. and Systems, Princeton University, March 1974.

[Neuse & Chandy 81]
Neuse, D., and K.M. Chandy. SCAT: a heuristic algorithm for queueing network models of computing systems. Proc. of the ACM/SIGMETRICS Conf. on Meas. and Modeling of Comp. Sys., Las Vegas, Sept. 1981, pp. 59-79.

[Noetzel 79]
Noetzel, A.S. A generalized queueing discipline for product form network solutions. *J. ACM* 26, 4 (Oct. 1979), 779-793.

[Pittel 76]
Pittel, B. Closed exponential networks of queues with blocking, the Jackson type stationary distribution and its asymptotic analysis. IBM Res. Rep. RC6174, Yorktown Heights, N.Y., 1976.

[Ramakrishnan & Mitra 82a]
Ramakrishnan, K.G., and D. Mitra. PANACEA user's manual for versions 1.0 and 1.1. Technical Memorandum 82-11211-5, Bell Laboratories, Feb. 1982.

[Ramakrishnan & Mitra 82b]
Ramakrishnan, K.G., and D. Mitra. An overview of PANACEA, a software package for analyzing Markovian queueing networks. *Bell System Tech. J.* 61, 10 (Dec. 1982), Part 1, 2849-2872.

[Reiser 81]
Reiser, M. Mean-value analysis and convolution method for queue-dependent servers in closed queueing networks. *Performance Evaluation* 1, 1 (Jan. 1981), 7-18.

[Reiser & Kobayashi 75]
Reiser, M., and H. Kobayashi. Queuing networks with multiple closed chains: theory and computational algorithms. *IBM J. Res. Develop.* 19, 3 (May 1975), 283-294.

[Reiser & Lavenberg 80]
Reiser, M., and S.S. Lavenberg. Mean value analysis of closed multichain queueing networks. *J. ACM* 27, 2 (April 1980), 313-322.

[Sauer 83]
Sauer, C.H. Computational algorithms for state-dependent queueing networks. *ACM Trans. on Comp. Sys.* 1, 1 (Feb. 1983), 67-92.

[Sauer & Chandy 81]
Sauer, C.H., and K.M. Chandy. *Computer System Performance Modeling*. Prentice-Hall, Englewood Cliffs, N.J., 1981.

[Schweitzer 79]
Schweitzer, P. Approximate analysis of multiclass closed networks of queues. Int. Conf. on Stochastic Control and Optimization, Amsterdam, 1979.

[Sevcik 77]
Sevcik, K.C. Priority scheduling disciplines in queueing network models of computer systems. Proc. IFIP 77, Toronto, Aug. 1977, pp. 565-570.

[Sevcik, Graham & Zahorjan 80]
Sevcik, K.C., G.S. Graham, and J. Zahorjan. Configuration and capacity planning in a distributed processing system. Proc. 16th. CPEUG, Orlando, Oct. 1980, pp. 165-171.

[Sevcik & Klawe 79]
Sevcik, K.C., and M.M. Klawe. Operational analysis versus stochastic modelling of computer systems. Proc. of Comp. Science and Statistics: 12th. Annual Symposium on the Interface, Waterloo, May 1979, pp. 177-184.

[Sevcik & Mitrani 81]
Sevcik, K.C., and I. Mitrani. The distribution of queuing network states at input and output instants. *J. ACM* 28, 2 (April 1981), 358-371.

[Suri 83]
Suri, R. Generalized Quick Bounds for performance of queueing networks. Tech. Rep. TR-05-83, Aiken Computation Laboratory, Harvard University, Cambridge, MA., Feb. 1983.

[THEsolver 81]
THEsolver user guide, Version 4.2. CSRG Tech. Note 18, University of Toronto, Toronto, Ont., Jan. 1981.

[Thomasian & Nadji 81]
Thomasian, A., and B. Nadji. Aggregation of stations in queueing network models of multiprogrammed computers. Proc. of the ACM/SIGMETRICS Conf. on Meas. and Modeling of Comp. Sys., Las Vegas, Sept. 1981, pp. 86-96.

[Towsley 80]
Towsley, D. Queuing network models with state-dependent routing. *J. ACM* 27, 2 (April 1980), 323-337.

[Zahorjan 80]
Zahorjan, J. The approximate solution of large queueing network models. Ph.D. Thesis, Tech. Rep. CSRG-122, University of Toronto, Toronto, Ont., Aug. 1980.

[Zahorjan et al. 82]
Zahorjan, J., K.C. Sevcik, D.L. Eager, and B.I. Galler. Balanced job bound analysis of queueing networks. *Comm. ACM* 25, 2 (Feb. 1982), 134-141.

[Zahorjan, Lazowska & Garner 83]
Zahorjan, J., E.D. Lazowska, and R.L. Garner. A decomposition approach to modelling high service time variability. *Perf. Eval* 3, 1 (Feb. 1983), 35-54.

[Zahorjan & Wong 81]
Zahorjan, J., and E. Wong. A solution of separable queueing network models using Mean Value Analysis. Proc. of the ACM/SIGMETRICS Conf. on Meas. and Modeling of Comp. Sys., Las Vegas, Sept. 1981, pp. 80-85.

University of Toronto
Computer Systems Research Group

BIBLIOGRAPHY OF CSRG TECHNICAL REPORTS 1980 - present

* + Out of print

*+CSRG-108 DIALOGUE ORGANIZATION AND STRUCTURE FOR INTERACTIVE INFORMATION SYSTEMS
John Leonard Barron
[M.Sc. Thesis, DCS, 1980]

*+CSRG-109 A UNIFYING MODEL OF PHYSICAL DATABASES
D.S. Batory, C.C. Gotlieb, April 1980

*+CSRG-110 OPTIMAL FILE DESIGNS AND REORGANIZATION POINTS
D.S. Batory, April 1980

*+CSRG-111 A PANACHE OF DBMS IDEAS III
D. Tsichritzis (ed.), April 1980

*+CSRG-112 TOPICS IN PSN - II: EXCEPTIONAL CONDITION HANDLING IN PSN; REPRESENTING PROGRAMS IN PSN; CONTENTS IN PSN
Yves Lesperance, Byran M. Kramer, Peter F. Schneider
April, 1980

CSRG-113 SYSTEM-ORIENTED MACRO-SCHEDULING
C.C. Gotlieb and A. Schonbach
May 1980

*+CSRG-114 A FRAMEWORK FOR VISUAL MOTION UNDERSTANDING
John Konstantine Tsotsos
[Ph.D. Thesis, DCS, June 1980]

*+CSRG-115 SPECIFICATION OF CONCURRENT EUCLID
James R. Cordy and Richard C. Holt
July 1980

CSRG-116 THE REPRESENTATION OF PROGRAMS IN THE PROCEDURAL SEMANTIC NETWORK FORMALISM
Bryan M. Kramer
[M.Sc. Thesis, DCS, 1980]

CSRG-117 CONTEXT-FREE GRAMMARS AND DERIVATION TREES AS PROGRAMMING TOOLS
Volker Linnemann
September 1980

+CSRG-118 S/SL: SYNTAX/SEMANTIC LANGUAGE INTRODUCTION AND SPECIFICATION
R.C. Holt, J.R. Cordy, D.B. Wortman
CSRG, September 1980

•+CSRG-119 PT: A PASCAL SUBSET
Alan Rosselet
[M.Sc. Thesis, DCS, October 1980]

CSRG-120 PTED: A STANDARD PASCAL TEXT EDITOR BASED ON THE KERNIGHAN AND PLAUGER DESIGN
Ken Newman, DCS
October 1980

CSRG-121 TERMINAL CONTEXT GRAMMARS
Howard W. Trickey
[M.Sc. Thesis, EE, September 1980]

CSRG-122 THE APPROXIMATE SOLUTION OF LARGE QUEUEING NETWORK MODELS
John Zahorjan
[Ph.D. Thesis, DCS, August 1980]

CSRG-123 A FORMAL TREATMENT OF IMPERFECT INFORMATION IN DATABASE MANAGEMENT
Yannis Vassiliou
[Ph.D. Thesis, DCS, September 1980]

CSRG-124 AN ANALYTIC MODEL OF PHYSICAL DATABASES
Don S. Batory
[Ph.D. Thesis, DCS, January 1981]

CSRG-125 MACHINE-INDEPENDENT CODE GENERATION
Richard H. Kozlak
[M.Sc. Thesis, DCS, January 1981]

CSRG-126 COMPUTER MACRO-SCHEDULING FOR HIGH PRODUCTIVITY
Abraham Schonbach
[Ph.D. Thesis, DCS, March 1981]

CSRG-127 OMEGA ALPHA
D. Tsichritzis (ed.), March 1981

+CSRG-128 DIALOGUE AND PROCESS DESIGN FOR INTERACTIVE INFORMATION SYSTEMS USING TAXIS
John Barron, April 1981

CSRG-129 DESIGN AND VERIFICATION OF INTERACTIVE INFORMATION SYSTEMS USING TAXIS
Harry K.T. Wong
[Ph.D. Thesis, DCS, to be submitted]

CSRG-130 DYNAMIC PROTECTION OF OBJECTS IN A COMPUTER UTILITY
Leslie H. Goldsmith, April, 1981

CSRG-131 INTEGRITY ANALYSIS: A METHODOLOGY FOR EDP AUDIT AND DATA QUALITY CONTROL
Maija Irene Svanks
[Ph.D. Thesis, DCS, February 1981]

CSRG-132 A PROTOTYPE KNOWLEDGE-BASED SYSTEM
FOR COMPUTER-ASSISTED MEDICAL DIAGNOSIS
Stephen A. Ho-Tai
[M.Sc.Thesis, DCS, January 1981]

*+CSRG-133 SPECIFICATION OF CONCURRENT EUCLID
James R. Cordy, Richard C. Holt
August 1981 (Version 1)

CSRG-134 ANOTHER LOOK AT COMMUNICATING PROCESSES
E.C.R. Hehner and C.A.R. Hoare, July, 1981

CSRG-135 ROBUST CONCURRENCY CONTROL IN DISTRIBUTED DATABASES
Derek L. Eager
[M.Sc. Thesis, DCS, October 1981]

CSRG-136 ESTIMATING SELECTIVITIES IN DATA BASES
Stavros Christodoulakis
[Ph.D. Thesis, DCS, December 1981]

CSRG-137 SATISFYING DATABASE STATES
Marc H. Graham
[Ph.D. Thesis, DCS, December 1981]

* CSRG-138 IMPROVING THE PERFORMANCE OF DATA BASE SYSTEMS
Geovane Cayres Magalhaes
[Ph.D. Thesis, DCS, December 1981]

+CSRG-139 A FORMAL TREATMENT OF INCOMPLETE KNOWLEDGE BASES
Hector J. Levesque
[Ph.D. Thesis, DCS, February 1982]

CSRG-140 AN OVERVIEW OF TUNIS: A UNIX LOOK-ALIKE
WRITTEN IN CONCURRENT EUCLID
R.C. Holt, February 1982

CSRG-141 ON PROVING THE ABSENCE OF EXECUTION ERRORS
W. David Elliott
[Ph.D. Thesis, DCS, September 1980]

+CSRG-142 A METHODOLOGY FOR PROGRAMMING WITH CONCURRENCY
Christian Lengauer
[Ph.D. Thesis, DCS, April 1982]

CSRG-143 ALPHA BETA
F. Lochovsky (ed.), May 1982

CSRG-144 A FIRST-ORDER DYNAMIC LOGIC FOR PLANNING
Henry A. Kautz
[M.Sc. Thesis, DCS, May 1982]

CSRG-145 ASYNCHRONOUS MULTIPLE ACCESS TREE ALGORITHMS
M.L. Molle, August 1982

CSRG-146 DETECTING INTERSECTION AMONG STAR POLYGONS
Delfin Montuno, Alain Fournier
September 1982

+CSRG-147 CONCURRENCY CONTROL PERFORMANCE ISSUES
Bruce I. Galler
[Ph.D. Thesis, DCS, September 1982]

CSRG-148 FINDING X-Y CONVEX HULL OF A SET OF X-Y POLYGONS
Delfin Y. Montuno, Alain Fournier
November 1982

CSRG-149 VIRTUAL TIME CSMA: A STUDY
Dimitri Konstantas
[M.Sc. Thesis, DCS, January 1983]

CSRG-150 BETA GAMMA
D. Tsichritzis (ed.) 1983

CSRG-151 A COMPUTATIONAL MODEL FOR THE ANALYSIS OF ARGUMENTS
Robin Cohen
[Ph.D. Thesis, DCS, October 1983]

CSRG-152 STRUCTURE OF A PORTABLE OPERATING SYSTEM
Mark P. Mendell
[M.Sc.Thesis, DCS, 1983]

CSRG-153 THE TURING LANGUAGE REPORT
Richard C. Holt and James R. Cordy
December 1983

CSRG-154 AN OBJECT-ORIENTED OFFICE DATA MODEL
Simon J. Gibbs
[Ph.D. Thesis, DCS, January 1984]

CSRG-155 REQUIREMENTS MODELING: A KNOWLEDGE REPRESENTATION APPROACH TO SOFTWARE REQUIREMENTS DEFINITION
Sol J. Greenspan
[Ph.D. Thesis, DCS, March 1984]

CSRG-156 BOUNDING ALGORITHMS FOR QUEUEING NETWORK MODELS OF COMPUTER SYSTEMS
Derek L. Eager
[Ph.D. Thesis, DCS, March 1984]